

VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA
EKONOMICKÁ FAKULTA

KATEDRA APLIKOVANÉ INFORMATIKY

Návrh a implementace databázové aplikace klientů pro zkušební laboratoř
Design and Implementation of the Clients' Database Application for the Testing
Laboratory

Student: Filip Horák
Vedoucí bakalářské práce: Ing. Vítězslav Novák, Ph.D.

Ostrava 2016

Zadání bakalářské práce

Student:

Filip Horák

Studijní program:

B6209 Systémové inženýrství a informatika

Studijní obor:

6209R017 Informatika v ekonomice

Téma:

Návrh a implementace databázové aplikace klientů pro zkušební
laboratoř

Design and Implementation of the Clients' Database Application for the
Testing Laboratory

Jazyk vypracování:

čeština

Zásady pro vypracování:

1. Úvod
 2. Teoretické pojmy a východiska tvorby databáze
 3. Analýza současného stavu ve firmě
 4. Návrh a realizace databázové aplikace
 5. Závěr
- Seznam použité literatury
Seznam zkratk
Prohlášení o využití výsledků bakalářské práce
Seznam příloh
Přílohy

Seznam doporučené odborné literatury:

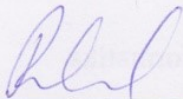
- KRUCZEK, Aleš. *Microsoft Access 2010: podrobná uživatelská příručka*. Brno: Computer Press, 2010.
ISBN 978-80-251-3289-0.
- PÍSEK, Slavoj. *Access 2010: podrobný průvodce*. Praha: Grada Publishing, 2011.
ISBN 978-80-247-3653-2.
- SHEPHERD, Richard. *Access VBA: výukový průvodce*. Brno: Computer Press, 2012.
ISBN 978-80-251-3686-7.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

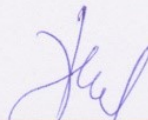
Vedoucí bakalářské práce: **Ing. Vítězslav Novák, Ph.D.**

Datum zadání: 20.11.2015

Datum odevzdání: 06.05.2016



Ing. Petr Rozehnal, Ph.D.
vedoucí katedry



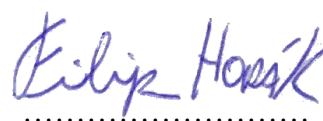
prof. Dr. Ing. Dana Dluhošová
děkanka fakulty

Místopřísežné prohlášení o samostatném vypracování bakalářské práce

„Prohlašuji, že jsem celou práci, včetně všech příloh, vypracoval samostatně.“

Zároveň bych na tomto místě chtěl poděkovat svému vedoucímu bakalářské práce Ing. Vítězslavu Novákovi, Ph.D. za vedení bakalářské práce, jeho cenné rady, připomínky a poskytnuté konzultace. Dále pak MUDr. Leopoldu Trdému za poskytnutí informací a materiálů potřebných k vypracování mé bakalářské práce.

Datum odevzdání bakalářské práce: 6. 5. 2016



Filip Horák

Obsah

1. Úvod.....	5
2. Teoretické pojmy a východiska tvorby databáze	7
2.1. Základní pojmy	7
2.1.1. Data.....	7
2.1.2. Informace.....	7
2.1.3. Pojem databáze	7
2.1.4. Systém řízení báze dat	8
2.1.5. Databázová aplikace	8
2.2. Databázové modely.....	8
2.2.1. Hierarchická databáze.....	9
2.2.2. Síťová databáze	9
2.2.3. Relační databáze	10
2.2.4. Objektová databáze.....	11
2.2.5. Objektově-relační databáze	11
2.3. Datové modelování	12
2.3.1. Konceptuální model.....	12
2.3.2. Logický model.....	15
2.3.3. Fyzický model	18
2.4. SQL.....	18
2.4.1. Data Definition Language (DDL)	19
2.4.2. Data Manipulation Language (DML).....	19
2.4.3. Data Control Language (DCL)	19
2.4.4. Transaction Control Language (TCL)	19
2.5. Microsoft Access 2010	20
2.5.1. Tabulky.....	21
2.5.2. Datové typy	22
2.5.3. Vlastnosti datových typů	23
2.5.4. Formuláře	24
2.5.5. Filtry	25
2.5.6. Dotazy.....	25

2.5.7.	Sestavy.....	27
2.5.8.	Makra.....	27
2.5.9.	Visual Basic for Application (VBA)	27
3.	Analýza současného stavu ve firmě	28
3.1.	O společnosti.....	28
3.2.	Současná situace	28
3.3.	Požadavky na aplikaci	29
4.	Návrh a realizace databázové aplikace.....	31
4.1.	Konceptuální návrh.....	31
4.1.1.	Přehled entit.....	32
4.2.	Logický návrh	36
4.3.	Aplikace Microsoft Access	37
4.3.1.	Tabulky.....	37
4.3.2.	Vztahy.....	43
4.3.3.	Formuláře	43
4.3.4.	Dotazy.....	51
4.3.5.	Sestavy.....	52
4.3.6.	Dokončení aplikace	53
4.4.	Implementace aplikace do firmy.....	54
5.	Závěr.....	55
	Seznam použité literatury	56
	Seznam zkratk.....	57
	Prohlášení o využití výsledků bakalářské práce	
	Seznam příloh	
	Přílohy	

1. Úvod

Většina podniků si potřebuje evidovat a uchovávat velké množství údajů, se kterými potřebuje dále pracovat. Vzhledem k neustálému vývoji v oblasti informačních technologií je žádoucí, aby si tyto údaje každý podnik jakéhokoliv zaměření a jakékoliv velikosti evidoval formou databázové aplikace.

Cílem této bakalářské práce je navrhnout a implementovat databázovou aplikaci klientů pro zkušební laboratoř v programovém prostředí Microsoft Access 2010. Smyslem aplikace je zefektivnit práci s veškerými údaji, které si potřebuje firma v databázi evidovat. Zkušební laboratoř, pro kterou je databázová aplikace tvořena, se zabývá prováděním zkoušek diagnostických zdravotnických rentgenových přístrojů a rentgenových pracovišť jako celků. Aplikace by měla sloužit pro shromažďování a uchovávání veškerých potřebných informací o klientech, provedení zkoušky jejich rentgenového přístroje, a dalších nezbytných údajů. S databází bude pracovat pouze jeden pracovník, který má za úkol domlouvat a koordinovat provádění zkoušek u klientů a měl by vycházet z informací uvedených v databázi a následně je upravovat do aktuální podoby. Konečná verze databázové aplikace by měla být pro pracovníka společnosti přehledná, intuitivní a pohodlně obsluhovatelná.

Bakalářská práce je členěna do pěti hlavních částí.

První částí práce je „Úvod“, ve kterém je popsán cíl bakalářské práce, uvedení do problematiky a členění hlavních částí včetně stručného popisu.

V další části s názvem „*Teoretické pojmy a východiska tvorby databáze*“ jsou vysvětleny základní teoretické pojmy potřebné pro porozumění termínu databázová aplikace. Součástí této kapitoly je taktéž seznámení s programem Microsoft Office 2010 a popis základních kamenů databáze pro její tvorbu.

Třetí část „*Analýza současného stavu ve firmě*“ je zaměřena na popis celé firmy a analýzu současného stavu v ní. Dále je specifikováno, co by měla finální databázová aplikace splňovat.

Klíčovou částí je kapitola „*Návrh a realizace databázové aplikace*“, která se zabývá základní praktickou částí celé bakalářské práce. Je zde popsán celý proces návrhu a tvorby databázové aplikace pomocí tabulek a jejich propojení relačními vztahy, vytvořenými formuláři, dotazy a sestavami, které by měly zajistit správný chod celé aplikace. Na konci této kapitoly je popsána implementace finální aplikace do firmy.

Konečnou částí bakalářské práce je „Závěr“, ve kterém je shrnutí průběhu návrhu databázové aplikace a zhodnocení, zda bylo či nebylo dosaženo stanovených cílů celé práce.

2. Teoretické pojmy a východiska tvorby databáze

Tato kapitola je zaměřena na teoretickou část bakalářské práce a slouží k objasnění a vysvětlení pojmů potřebných k porozumění a tvorbě databázové aplikace. Obsahem jsou základní informace spojené s databázemi a v neposlední řadě seznámení s programem Microsoft Access 2010, ve kterém je tvořen praktický výstup této bakalářské práce.

2.1. Základní pojmy

V této podkapitole jsou uvedeny elementární teoretické pojmy, které jsou zásadní pro základní pochopení a uvedení do dané problematiky. Konkrétně se jedná o pojmy Data, Informace, Databáze, Systém řízení báze dat a Databázová aplikace.

2.1.1. Data

Data jsou údaje mající například pro určitý podnikatelský subjekt daný význam. Základním rysem dat je, že neprošla žádným zpracováním a měla by sloužit jako podklad pro získávání relevantních informací. V databázové aplikaci se jedná o veškeré údaje uložené v tabulkách. (Conolly, Begg a Holowczak, 2009)

2.1.2. Informace

Informace jsou již nějakým požadovaným způsobem zpracovaná data mající konkrétní strukturu a přiřazený význam. Mají již faktický smysl pro uživatele a mohou být neomezeně prezentována. V praxi se může jednat například o výstup ve formuláři, sestavě, apod. (Conolly, Begg a Holowczak, 2009)

Jak tvrdí Hernandez (2006, s. 67) „Data jsou to, co ukládáme, informace jsou to, co získáváme.“

2.1.3. Pojem databáze

Pojem databáze je možné definovat vícero způsoby, ale obecně lze v počítačovém světě za databázi označit téměř vše, co obsahuje určitým způsobem uložené a utříděné informace. (Kruczek, 2010)

Oppel (2006) databázi definuje jako množinu vzájemně uspořádaných a souvisejících dat, která se jeví jako ucelená jednotka uložena na určitém paměťovém médiu.

Databáze je sdílená kolekce logicky souvisejících dat i s popisem své datové struktury, organizovaná pro optimální manipulaci s perzistentními daty a získávání informací pro potřeby informačního systému. (Novák, 2014)

Databázi si lze představit jako soubor dat sloužící pro popis reálného světa. Pojem databáze si však není nutné hned spojovat s informačními technologiemi, i běžná kartotéka je určitou formou databáze. Ovšem taková forma shlukování a ukládání dat má velké množství nevýhod. Může zde dojít například k redundanci a nekonzistenci dat, dalším problémem je zabezpečení dat, sdílení dat a v neposlední řadě obtížná správa dat. Proto se postupem času vyvinula možnost ukládat data v digitalizované podobě. S takovou databází se pak daleko praktičtěji a efektivněji pracuje. (Oppel, 2006)

2.1.4. Systém řízení báze dat

Systém řízení báze dat (SŘBD) z anglického Database Management System (DBMS) je software představující prostředníka mezi uloženými daty a aplikačními programy. Umožňuje uživatelům vytvářet, upravovat a spravovat databáze. (Connolly, Begg a Holowczak, 2009)

Pomocí systému řízení báze dat se zajišťuje efektivní práci s velkým množstvím dat a je umožněno je vkládat, udržovat, aktualizovat a mazat. Dalšími charakteristickými vlastnostmi je například využití jazyka vyšší úrovně (např. SQL), autentizace uživatelů a jejich autorizace, kontrola integrity dat, zálohování databáze, řízení paralelního přístupu k datům více uživatelům bez kolize, a mnoho dalších. (Oppel, 2008)

Existuje velké množství různých systémů řízení báze dat. Nejpoužívanějšími produkty SŘBD jsou Oracle Database společnosti Oracle Corporation, DB2 společnosti IBM, Microsoft SQL Server, Microsoft Access, MySQL, FireBird, a mnoho dalších.

2.1.5. Databázová aplikace

Databázová aplikace je softwarové vybavení fungující jako rozhraní spojující SŘBD a uživatele. Aplikace je naprogramována v některém z vyšších programovacích jazyků (Java, C++, VB, apod.). Součástí databázové aplikace jsou na rozdíl od databází i formuláře a sestavy, s nimiž může uživatel manipulovat pomocí požadavků zasílaných nejčastěji ve formě SQL příkazů. (Connolly, Begg a Holowczak, 2009)

2.2. Databázové modely

Pomocí databázového modelu se modeluje struktura databáze. Jedná se o architekturu, podle které je databázovému systému umožněno ukládat a propojovat jednotlivé objekty databáze. Existuje vícero typů databázových modelů a některé z nich jsou níže popsány. (Oppel, 2006)

2.2.1. Hierarchická databáze

Hierarchický model je nejstarší model, pomocí kterého se tvořily první databáze. Tento model byl vyvinut ze souborových systémů, které nahradil. Jednotlivé záznamy jsou uspořádány v hierarchii, podobně jako organizační schéma, a jsou propojeny pomocí ukazatelů obsahující adresu příslušného záznamu, podle které se dá určit, kde je záznam fyzicky uložen. Prostřednictvím ukazatele se zavádí mezi záznamy vztah rodiče a potomka, kde jeden rodič může mít více potomků, ale potomek může mít jen jednoho rodiče. Je nutné, aby měl uživatel dobré povědomí o struktuře dané databáze, jelikož přístup k hledaným datům probíhá přes kořenovou tabulku postupným procházením jejích potomků. Nejrozšířenější hierarchickou databází byla IMS od spol. IBM. (Oppel, 2006)

Výhodou hierarchického databázového modelu je rychlý přístup k vyhledávaným datům, čehož je docíleno explicitním propojením tabulek. Další výhodou je referenční integrita, pomocí které je zajištěno propojení záznamů v nadřazené a podřazené tabulce. Při přidání nebo změně záznamu v podřazené tabulce, je provedena kontrola, zda existuje záznam v nadřazené tabulce, pokud tomu tak není, dojde k chybě. Jelikož tento model podporuje pouze vztah 1:N může nastat problém s redundantními daty, což jsou data vyskytující se v databázi vícekrát. (Hernandez, 2006)

2.2.2. Síťová databáze

Síťový databázový model byl vyvinut přibližně ve stejnou dobu jako hierarchický model. Jedná se vlastně o zdokonalenou koncepci hierarchického modelu. Namísto terminologie rodič a potomek byly u tohoto modelu zavedeny pojmy vlastník a člen. V souvislosti s tímto modelem jsou i pojmy uzel, který vyjadřuje soubor záznamů a množinová struktura vyjadřující vztahy mezi uzly. Uzel vlastník může existovat bez vztahu s uzlem člen, ale člen nemůže existovat bez vazby na uzel vlastník. Příbuzné záznamy jsou také jako u hierarchického modelu propojeny prostřednictvím ukazatelů, ovšem v síťovém modelu jsou vztahy mezi záznamy pojmenovány a na základě toho se může člen účastnit vícero vztahů. (Oppel, 2006)

Za hlavní výhodu síťového modelu je označována rychlost vyhledávání dat spočívající v možnosti procházení databáze prostřednictvím množinových struktur. Je zde i možnost vytvářet komplexní dotazy, což nebylo v takové míře možné v hierarchickém modelu. Aby bylo možné efektivně pracovat s databází, je nutné znát její strukturu, což je považováno za největší nevýhodu. Problém může nastat i při potřebě změnit strukturu databáze a tím

zároveň poupravit její množinovou strukturu, jelikož je pak nutné změnit aplikační programy používající tyto množinové struktury. (Hernandez, 2006)

Síťový databázový model poskytuje větší flexibilitu, ovšem na úkor větší složitosti. Přílišná složitost a náročnost na údržbu byly hlavními důvody, proč tento model zanikl. Nejpopulárnější databází založenou na síťovém modelu byla IDMS vyvinutá společností Cullinane, později přejmenovanou na Cullinet. (Oppel, 2006)

2.2.3. Relační databáze

Předchozí dva databázové modely jsou neflexibilní, případně příliš složité. Vyhledávání v nich není příliš optimální, jelikož se musí složitě procházet celá databáze. Zlomový moment v rámci databázových modelů nastal, když americký počítačový vědec a výzkumník společnosti IBM Dr. E. F. Codd objevil relační databázový model a oficiálně ho představil v roce 1970 ve své práci s názvem „*Relační model dat pro velké sdílené databanky*“. (Oppel, 2006)

Relační databáze je konstruována tak, že se jejím prostřednictvím pracuje s množinou dat na rozdíl od síťového či hierarchického databázového modelu, kde se pracovalo s jednotlivými záznamy. Pomocí relačního databázového modelu je umožněno zpracovávat velké množství dat, která jsou ukládána v dvourozměrných tabulkách uspořádaných ze záznamů a atributů. Skutečné uspořádání atributů a záznamů je nepodstatné a uživatel nemusí znát fyzické umístění záznamu při získávání dat. Jednotlivé záznamy nejsou propojovány pomocí fyzických ukazatelů, jako tomu bylo u hierarchického a síťového databázového modelu, ale prostřednictvím vloženého společného atributu do jednotlivých tabulek, tzv. cizího klíče, který definuje vztah mezi dvěma tabulkami. Primární klíč je jednoznačný identifikátor záznamu a každá tabulka má mít právě jeden definovaný primární klíč. (Oppel, 2006), (Hernandez, 2006)

Databázi lze vytvářet, upravovat a celkově spravovat prostřednictvím jazyka SQL, který je důkladněji popsán v části 2.4 této bakalářské práce. Ovšem prostřednictvím grafických editorů, které již mají jazyk SQL zabudován, lze SQL příkazy vytvářet i bez znalosti tohoto jazyka. Relační model má tu výhodu, že lze poměrně snadno upravovat strukturu databáze, víceúrovňová integrita zajišťující jedinečnost záznamů a detekující chybějící hodnoty primárních klíčů je implementována automaticky a vyhledávání dat z jedné nebo více tabulek ve vzájemném vztahu je vcelku jednoduché. (Hernandez, 2006)

Relační databázový model byl velmi kladně přijat a to hlavně díky schopnosti zachytit reálné situace a své flexibilitě a stal se z něj základ většiny databázových systémů a je pravděpodobné, že tomu tak ještě mnoho dalších let bude. (Oppel, 2006)

2.2.4. Objektová databáze

Počátek objektově-orientovaného databázového modelu se datuje do 70. let minulého století, ale až do roku 1990 nepředstavoval větší komerční využití. Tento model se začal využívat zejména proto, že pomocí tehdejších používaných modelů nebylo možné pracovat s náročnějšími daty, jako jsou obrázky, audio či video soubory, apod.

Objektem se rozumí logické seskupení příbuzných dat a programové logiky, které představuje skutečnou věc reálného světa, například objednávku ve firmě. Jednotlivé datové položky, jako například číslo objednávky, se nazývají proměnné a jsou uloženy uvnitř každého objektu. S objektově-orientovaným modelem je spjat i pojem metoda, což je část aplikační logiky fungující nad určitým objektem a poskytuje konečnou funkci. Existuje mnoho rozdílů mezi objektově-orientovaným databázovým modelem a modely zmíněnými dříve v této bakalářské práci, ale nejvýznamnějším rozdílem je fakt, že jednotlivé proměnné mohou být přístupné pouze prostřednictvím metod. Tato vlastnost se nazývá zapouzdření. Objekty jsou uspořádány do hierarchie tříd, ve kterých jsou nadefinované jednotlivé proměnné a metody. Společné metody a definice proměnných stačí definovat pouze jednou a poté jsou dostupné dalším členům třídy, a to díky další důležité vlastnosti, které se říká dědičnost. (Oppel, 2006)

Tento databázový model je postaven na objektově-orientovaných programovacích jazycích, jako jsou např. Java, C++, PHP, a nese charakteristiky a výhody těchto jazyků. Stal se velice oblíbeným a používaným modelem, a to i díky snadné a efektivní úpravě databáze pomocí programového kódu. (Šimonová a Panuš, 2007)

2.2.5. Objektově-relační databáze

Někteří distributoři relačních databází vzali na vědomí významný přínos objektově-orientovaného modelu a rozhodli se zkusit propojit to nejlepší z těchto dvou modelů do jednoho nového databázového modelu. Původně byl tento model pojmenován jako univerzální databáze, ale nakonec se dala přednost techničtěji přesnějšímu názvu objektově-relační databáze. K plnému pochopení toho modelu je nutná podrobnější znalost relačního i objektově-orientovaného modelu. Nejznámějšími objektově-relačními modely na trhu jsou Oracle, DB2 od společnosti IBM a Informix. (Oppel, 2006)

2.3. Datové modelování

Pojem datové modelování značí proces, při kterém se definují a analyzují požadavky na strukturu dat s určitou mírou abstrakce a s rozdílným pohledem na data v jednotlivých fázích. Cílem je zachytit a popsat část reality a převést tyto reálné objekty na datové objekty. Základním principem modelování je prvotní vytvoření modelu daného systému a až posléze vytvoření samotného systému. Při vytváření modelu se zjednodušuje a abstrahuje realita od méně podstatných skutečností a to z důvodu, že není možné realitu modelovat do všech detailů. Model by měl obsahovat jen nejpodstatnější a nejdůležitější skutečnosti. (Conolly, Begg a Holowczak, 2009), (Šimonová a Panuš, 2007)

Datové modelování zahrnuje postupně tři fáze vývoje, ve kterých se pohlíží na data odlišným způsobem a v každé fázi je vytvořen jiný typ modelu, které jsou popsány níže.

2.3.1. Konceptuální model

Jedná se o model první fáze datového modelování. Jsou v něm využity data používané v konkrétních organizacích s určitou mírou abstrahování detailů, jako je pozdější fyzická implementace, nebo nad jakou technologií bude výsledný systém nakonec vytvořen. Pomocí konceptuálního modelu se identifikují klíčové entity a jejich vztahy popisující situace reálného světa, které je potřeba přenést do databáze. Model slouží jako podklad logického modelu. (Conolly, Begg a Holowczak, 2009), (Šimonová a Panuš, 2007)

K popisu databázového systému se většinou používá grafická metoda vytvoření entitně-relačního diagramu, který umožňuje kompletní návrh databáze. Poskytuje podklad pro návrh fyzické struktury souborů a používá se zejména k vytvoření konceptuálního a logického modelu. Součástí ERD jsou entity a jejich vzájemné vazby, atributy těchto entit, domény a klíče. (Conolly, Begg a Holowczak, 2009)

2.3.1.1. Entita

Pojmem entita se rozumí jakýkoliv objekt z reálného světa, který má pro firmu určitý význam a jsou o něm shromažďovány informace, které jsou následně zaznamenávány do databáze. Může se jednat o osobu, věc nebo místo, důležité je vybírat vhodné entity, aby byly zajištěny potřeby uživatele databáze a její správná funkčnost. Každá entita nacházející se v konceptuálním modelu reprezentuje celou svou třídu a jednotlivé výskyty dané entity jsou označeny jako její instance. V praxi se může jednat například o „Výrobek“, který je entitou a obsahuje své atributy a instancí této entity specifikující již nějaký konkrétní výrobek. (Oppel, 2006)

„Množina objektů se shodnými vlastnostmi, které uživatel nebo organizace identifikuje jako nezávislé existující objekty.“ (Conolly, 2009, s. 156)

2.3.1.2. Atribut

Slouží k charakterizování či popisu dané entity. Hodnotu atributu by nemělo být možné již rozdělit na další menší jednotky, tj. měl by být tzv. atomický. Měl by být nejmenší možnou pojmenovatelnou jednotkou dat definovanou v databázovém systému. Například u entity „Auto“ můžeme jako atributy definovat značku auta, typ, rok výroby, barvu, apod. Některé atributy mohou tvořit jednoznačný identifikátor entity obsahující specifickou hodnotu pro každý výskyt této entity, jedná se o tzv. kandidátní klíč. (Oppel, 2006)

2.3.1.3. Domény

Doména je rozsah hodnot, kterých může atribut nabývat. Může se například explicitně nastavit u atributu rok výroby rozmezí od 1900 do 2016. (Šimonová a Panuš, 2007)

2.3.1.4. Vztahy

Vztahy definují propojení vzájemně souvisejících dat uložených v rozdílných entitách. Při navrhování konceptuálního modelu je nutno pochopit jednotlivé vztahy mezi entitami, aby byla zaručena správná funkcionalita. Pomocí vztahů je umožněno redukovat nadbytečná data a eliminovat data duplicitní. Každý vztah má tři charakteristické vlastnosti, kterými jsou kardinalita, stupeň a volitelnost. (Oppel, 2006), (Hernandez, 2006)

Kardinalita, může být označována také jako typ vztahu, představuje počet výskytů entity v určitém vztahu. Dělí se na tři typy vztahů, a to 1:1, 1:N a M:N. Není-li mezi atributy v entitách žádná spojitost, nedefinuje se žádný vztah.

- 1) **Vztah 1:1** – Jedná se o takový vztah, ve kterém lze jednomu záznamu, neboli instanci entity, přiřadit maximálně jednu instanci jiné entity. Ze všech tří typů relačních vztahů je tento typ používán nejméně často, jelikož nepřináší příliš výhod a většinou lze obě tyto entity nahradit pouze jednou. Používá se ve speciálních případech, například když je potřeba zabezpečit záznam a v důsledku toho ho rozdělit do více tabulek, nebo z důvodu nepřehlednosti tabulek, když je počet sloupců příliš velký. (Oppel, 2006), (Kruczek, 2010)
- 2) **Vztah 1:N** – Vztahy 1:N jsou nejspíše nejběžněji používané. Každý záznam na jedné straně může odpovídat více záznamům na straně druhé, tedy na straně N. V praxi se může jednat například o databázi dodavatelů. S každým dodavatelem může souviset více faktur, ale faktura je vázaná pouze k jednomu dodavateli.

- 3) **Vztah M:N** – Tento typ vztahu znamená, že ke každé instanci entity na straně M se může přiřadit více instancí entity se shodným klíčem na straně N a naopak. Například na jedné straně entita Film a na druhé straně entita Herec. Ve filmu hraje několik herců, a zároveň jednotliví herci mohou hrát ve více filmech. Pomocí konceptuálního modelu lze tento vztah bez jakýchkoliv problémů zobrazit, ovšem při fyzickém řešení je nutné ho převést na vztah typu 1:N. Vytvoří se nová tabulka obsahující klíče ze dvou původních tabulek a s každou z nich se propojí vztahem typu 1:N. Vztah M:N je tedy pomocí nové tabulky převeden na dva vztahy typu 1:N. (Kruczek, 2010)

Pomocí stupně účasti se určuje minimální počet instancí dané entity, které musí být ve vztahu k jedné instanci druhé entity a maximální počet instancí dané entity, které mohou mít vztah k jedné instanci druhé entity. Existují vztahy unární, binární, ternární až N-ární. Unární vztah značí, že je entita ve vztahu sama se sebou. Druhý stupeň účasti, tzv. binární vztah, představuje vzájemný vztah mezi dvěma entitami. Poslední dva zmíněné se v praxi téměř neobjevují a to z důvodu nesnadného modelování. (Hernandez, 2006), (Kroenke a Auer, 2015)

Třetí charakteristikou vztahů je tzv. volitelnost, která se dále dělí na povinnou a volitelnou. Určuje, zda musí existovat záznam v druhé tabulce dříve, než je vložen záznam do první tabulky. Povinný typ účasti udává, že než je vložen záznam do druhé tabulky je nutné vložit záznam do první tabulky. U volitelné účasti není nutné do první tabulky vkládat jakékoliv záznamy, aby mohl být do druhé tabulky záznam vložen. (Hernandez, 2006)

2.3.1.5. Klíče

Klíče jsou atributy, které slouží k jednoznačnému identifikování každého záznamu v tabulce. Může se jednat o jeden atribut, případně o kombinaci atributů. Existují různé typy klíčů a jsou níže jednotlivě popsány (Conolly, Begg a Holowczak, 2009)

- 1) **Superklíč** – Jedná se o atribut či množinu atributů jedinečně identifikující každý výskyt entity. Může obsahovat nadbytečné atributy nepotřebné pro identifikaci instance entity. (Conolly, Begg a Holowczak, 2009)
- 2) **Kandidátní klíč** – Atribut či kombinace atributů jednoznačně identifikující instance entity. Z kandidátních klíčů se může určit primární klíč, a ostatní budou alternativními klíči. V rámci jednoho záznamu nemůže existovat více stejných

kandidátních klíčů. Pokud se vynechá jakákoliv část kandidátního klíče, není zaručena jeho jedinečnost. (Conolly, Begg a Holowczak, 2009)

- 3) **Primární klíč** – Jednoznačně identifikuje každý záznam tabulky a může být tvořen jedním, případně více sloupci tabulky. Primární klíč musí vždy mít jedinečný index bez duplicit, což je nastaveno implicitně. Indexace je popsána v kapitole 2.5 této bakalářské práce. Pokud je potřeba tvořit pomocí tabulky vztahy na jiné tabulky, tak je nutné, aby byl primární klíč určen. V opačném případě primární klíč není potřeba, ovšem je doporučeno každé tabulce ho přiřazovat. (Kruczek, 2010)
- 4) **Cizí klíč** – Cizí klíč je atribut či skupina atributů odkazující na primární klíč v jiné entitě. Jedná se o kopii primárního klíče začleněnou do tabulky na straně N, ve vztahu 1:N. Je používán k určení vztahu mezi dvěma tabulkami. (Kruczek, 2010)
- 5) **Alternativní klíč** – Alternativní klíč je kandidátní klíč, který není primárním klíčem příslušné entity. (Conolly, Begg a Holowczak, 2009)

2.3.1.6. Postup tvorby konceptuálního modelu

V prvním kroku tvorby konceptuálního modelu se graficky vyznačí struktura entit a jednotlivě se pojmenují. Posléze se ke všem entitám přiřadí příslušné atributy. Ve třetím kroku se určují vztahy mezi entitami, jejich kardinalita a volitelnost. V poslední kroku se integrují dílčí části modelu. Důležitý je tento krok především u rozsáhlých projektů, kdy je třeba integrovat do jednoho konceptuálního modelu několik dílčích konceptuálních modelů. Nejrozšířenějším způsobem integrace je binární přístup, pomocí kterého se postupně integrují dvojice dílčích modelů. (Kaluža a Kalužová, 2012)

2.3.2. Logický model

Vytvoření logického modelu je druhým krokem datového modelování. Tvorba je prováděna na základě výstupu z konceptuálního modelu. Představuje střední míru abstrakce a určuje se v této fázi modelování, jak budou data strukturována, tj. například zda budou uchovávána v objektové databázi nebo relační, apod. Dále v této části je popsán relační databázový model, jelikož je reálně použit k této bakalářské práci a zároveň se jedná o nejpoužívanější logický model. (Conolly, Begg a Holowczak, 2009)

Částečně je relační databázový model popsán již v kapitole 2.2.3 této bakalářské práce a pojem vztah ve čtvrté podkapitole kapitoly 2.3.1. Zde jsou dále popsány další pojmy související s relačním databázovým modelem.

2.3.2.1. Tabulka

Tabulka je základem každé relační databáze a skládá se ze sloupců reprezentujících jednotlivé atributy a z řádků, které reprezentují jednotlivé záznamy. Z konceptuálního modelu jsou entity převedeny do tabulek logického modelu. Každá tabulka většinou reprezentuje jednu entitu, ale může se stát, že jedna tabulka bude reprezentovat několik entit, případně jednu entitu několik tabulek. (Oppel, 2006)

Tabulka v relační databázi má nejružnější vlastnosti.

- Název tabulky ji jednoznačně identifikuje a je jedinečný v celé databázi.
- Sloupce v rámci tabulky jsou pojmenovány jednoznačně a jedinečně.
- Tabulka neobsahuje duplicitní záznamy, díky čemuž je docíleno správného zavedení primárního klíče.
- Pořadí sloupců ani řádku nehraje roli. (Conolly, Begg a Holowczak, 2009)

2.3.2.2. Normalizace

Pojem normalizace označuje proces rozkládání velké tabulky na menší, aby se zamezilo výskytu redundantních dat, zjednodušila se práce a manipulace s daty a zaručila se lepší konzistence dat. Rozložené tabulky by mělo být možné zpětně zase spojit do původní tabulky bez ztráty informace, tento princip je nazýván bezztrátová dekompozice. Pomocí normalizace by mělo být dosaženo odstranění anomálií z modelu, které se dále dělí na anomálie při vkládání, anomálie při odstraňování a anomálie při aktualizaci. (Conolly, Begg a Holowczak, 2009), (Kulhan, 2008)

Pravidla, která by měla data v tabulce splňovat, jsou nazvány normální formy. Postupuje se postupně od nižších forem k vyšším a každá vyšší forma zahrnuje všechny nižší. Obecně je definováno šest normálních forem, a to 1NF, 2NF, 3NF, BCNF, 4NF a 5NF. Existuje i 0NF, tedy nultá forma, ve které se tabulka vyskytne, nejsou-li data v nějakém poli atomická, což je v odporu s relačním modelem, ve kterém data v jednom poli nesmí být dále dělitelná. (Kulhan, 2008)

- 1) **První normální forma (1NF)** – Tabulka je v této formě, pokud v ní nejsou opakující se záznamy, data jsou atomická a jsou definovány veškeré klíčové atributy včetně primárního klíče.
- 2) **Druhá normální forma (2NF)** – Tato forma vychází z 1NF a zároveň je každý neklíčový atribut v dané tabulce závislý na primárním klíči. Jestliže primární klíč

zahrnuje pouze jedno pole, tak je tabulka automaticky v druhé normální formě, ovšem když splňuje i první normální formu.

- 3) **Třetí normální forma (3NF)** – Tabulka je ve třetí normální formě, pokud je ve druhé normální formě a zároveň není žádný její atribut tranzitivně závislý na primárním klíči. Tranzitivní závislost značí závislost mezi alespoň dvěma atributy a klíčem a platí, že jeden atribut je funkčně závislý na klíči a druhý na prvním atributu. Platí tedy, že všechny neklíčové atributy musí být navzájem nezávislé.
- 4) **Boyce Coddova normální forma (BCNF)** – Jedná se o vylepšenou variantu třetí normální formy. Pomocí této normální formy je řečeno, že nesmí být žádná závislost mezi kandidátními klíči. Každá tabulka, která je v této formě, je zároveň i ve třetí normální formě. Naopak tomu tak nemusí být, jestliže platí, že v tabulce existuje více kandidátních klíčů, veškeré kandidátní klíče jsou složeny z více než jednoho atributu, nebo jestli existuje společný atribut pro všechny kandidátní klíče.
- 5) **Čtvrtá normální forma (4NF)** – Tabulka je v této normální formě, když je v BCNF a prostřednictvím tabulky se popisuje pouze příčinná souvislost mezi klíčem a atributy. Je vyžadováno, aby primární klíč byl tvořen jen hodnotami, které mají skutečnou vzájemnou souvislost.
- 6) **Pátá normální forma (5NF)** – V páté normální formě je tabulka, pokud je ve čtvrté normální formě a tabulku již nelze bezztrátově rozložit. (Novák, 2014)

2.3.2.3. Relační algebra

Relační algebra slouží pro manipulaci s tabulkami a používá se zejména na popis operací v relačních databázích. Pomocí relační algebry je umožněno pracovat s daty v tabulkách. Operandů neboli vstupní hodnotou algebry jsou tabulky. Operátory se dělí na tradiční nazývané též množinové, mezi které patří sjednocení, rozdíl, součin a průnik a na speciální, do kterých patří selekce, dělení, spojení a projekce. (Šimonová a Panuš, 2007)

2.3.2.4. Omezení

V rámci databáze lze nastavit různá omezení týkající se databázových objektů, konkrétně povětšinou tabulek. Tato omezení nastavují zejména administrátoři. Pomocí omezení se definují hodnoty, které je možné do tabulek zadat.

- **Omezení primárního klíče** – Sloupec případně více sloupců tabulky nastavených jako primární klíč slouží k zajištění jedinečnosti každého záznamu.

- **Referenční omezení** – Pomocí referenčního omezení, neboli referenční integrity se kontrolují a zajišťují vztahy mezi tabulkami. Cizí klíč v dané tabulce musí mít hodnotu primárního klíče z tabulky, ve které je definován.
- **Omezení integrity** – Toto omezení slouží ke zvýšení kvality dat. Jedná se například o omezení CHECK, což je ověřovací pravidlo vracející hodnotu pravda či nepravda a platí, že záznam je možné do tabulky zadat, pokud vrátí hodnotu pravda. Dalším omezením integrity je omezení NOT NULL, pomocí kterého se zabezpečuje, že zadaná hodnota nesmí být nulová. (Oppel, 2006)

2.3.2.5. Postup tvorby relačního modelu

Relační modelování přetváří konceptuální model ve formě ER diagramu do logického modelu. Prvním krokem je vytvoření soustavy předběžných tabulek určených na základě entit konceptuálního modelu. Tabulkám jsou přiřazeny prozatím pouze jména, primární, cizí a kandidátní klíče. Dalším krokem je přiřazení zbývajících atributů. Ve třetím kroku se provádí tzv. normalizace modelu, což je proces, pomocí kterého se odstraňují anomálie navrženého modelu. Posléze se provede revize konceptuálního modelu z důvodu jeho úprav. Pátým a zároveň posledním krokem je specifikace domén, ve které se atributům přiřadí jejich charakteristiky. (Kaluža a Kalužová, 2012)

2.3.3. Fyzický model

Vytvoření fyzického modelu je poslední krokem datového modelování. Vychází z výstupů logického modelu a jedná se již o realizaci databázové aplikace v konkrétním softwarovém vybavení, jako je například Oracle, SQL Server nebo DB2. V této bakalářské práci je využit software Microsoft Access, který je detailněji popsán v kapitole 2.5.

2.4. SQL

SQL je univerzální strukturovaný programovací jazyk používající se k práci s relačními databázemi. Slouží k vytváření struktury jednotlivých databází a tabulek a k organizování, správě a získávání dat, které jsou uloženy v databázi. K datům přistupuje prostřednictvím zasílání požadavků respektive dotazů dané databázi, ovšem neurčuje, jak se toho má dosáhnout. Databáze na základě těchto dotazů zašle zpět svou odpověď. Výhodou tohoto jazyka je jeho přenositelnost mezi různými platformami.

Příkazy SQL jazyka se dělí do několika kategorií na základě jejich funkcí. Pro všechny shodně platí, že mají stejnou syntaxi a musí se u nich dodržovat stejná pravidla. Jednotlivé kategorie jsou popsány níže v podkapitolách této kapitoly. (Oppel, 2008)

2.4.1. Data Definition Language (DDL)

DDL, neboli jazyk pro definici dat obsahuje příkazy sloužící k vytváření, mazání a upravování databázových objektů. Tyto příkazy neslouží k vytváření nebo upravování dat v tabulkách, ale k tvorbě struktury databáze. Konkrétně se jedná o příkazy:

- CREATE – tento příkaz slouží k vytváření nových databázových objektů,
- ALTER – provádějí se pomocí něj změny již existujících objektů,
- DROP – slouží k odstraňování nepotřebných objektů. (Oppel, 2008)

2.4.2. Data Manipulation Language (DML)

Data Manipulation Language, respektive v českém překladu jazyk pro manipulaci s daty, konkrétně slouží k získávání dat z databáze a k jejich úpravě. Pomocí příkazů DML jazyka lze manipulovat s daty tabulek a mezi takové příkazy patří například:

- SELECT – pomocí tohoto příkazů se vybírají data z databáze a odpověď na tento příkaz je ve tvaru tabulky, lze vybírat data i z více tabulek,
- INSERT – tento příkaz slouží k přidávání nových záznamů do tabulky,
- UPDATE – slouží k upravování již stávajících záznamů v dané tabulce,
- DELETE – pomocí tohoto příkazu je umožněno mazat nepotřebné záznamy,
- MERGE – buďto se data do tabulky vloží, a to pokud neexistuje odpovídající klíč, případně když klíč existuje, tak se data v tabulce upraví, jedná se vlastně o kombinaci příkazů INSERT a UPDATE. (Oppel, 2008)

2.4.3. Data Control Language (DCL)

Příkazy tohoto jazyka slouží ke správě dat a souvisí také se zabezpečením databáze. DCL jazyk slouží především administrátorům databází, kteří pomocí něj přidělují oprávnění v rámci konkrétního systému řízení báze dat. Mezi takové příkazy patří:

- CREATE USER – příkaz je určený k vytvoření nového uživatele,
- DROP USER – pomocí tohoto příkazu se odstraňuje uživatel,
- GRANT – příkaz slouží k přidělení oprávnění uživatelům v rámci databáze,
- REVOKE – slouží k odebrání oprávnění uživateli. (Oppel, 2008)

2.4.4. Transaction Control Language (TCL)

Příkazy tohoto jazyka slouží pro řízení databázové transakce, což je uspořádaná množina prvků jevící se jako ucelená jednotka. Jedná se například o příkazy:

- START TRANSACTION – slouží k zahájení databázové transakce,

- COMMIT – jedná se o příkaz určený k potvrzení transakce,
- ROLLBACK – zruší se transakce a databáze se vrátí do původního stavu,
- SAVEPOINT – slouží k definování záchytného bodu, aby se například při chybě mohla transakce vrátit zpět jen částečně. (Oppel, 2008)

Každá transakce musí splňovat 4 základní vlastnosti pod zkratkou ACID.

- **Atomicita** – transakce musí být provedena jako celek a při úspěšném zakončení transakce jsou provedeny změny v databázi, v případě, že se transakce celá úspěšně neprovede, vrátí se databáze do původního stavu.
- **Konzistence** – pomocí transakce se převádí stav databáze z jednoho konzistentního stavu do druhého konzistentního stavu.
- **Izolace** – tato vlastnost definuje, že žádná transakce by neměla být závislá na jiné transakci, i když se tyto transakce provádějí současně.
- **Trvanlivost** – každá změna v databázi provedena prostřednictvím transakce by měla být trvalá i při vypnutí či havárii databáze. (Oppel, 2008)

2.5. Microsoft Access 2010

Databázová aplikace k této bakalářské práci je vytvořena v programu Microsoft Access 2010 programového balíku Microsoft Office 2010 edice Professional, a proto je tato kapitola podrobněji věnována tomuto programu.

Jedná se o aplikaci určenou pro práci s relační databází a její správu. Microsoft Access propojuje systém řízení báze dat s grafickým uživatelským rozhraním. Prostřednictvím tohoto programového vybavení je umožněno:

- Data uchovávat,
- pracovat s daty, jako je zobrazovat, upravovat, přidávat nové záznamy a staré mazat, apod.,
- automatizovat manipulaci s daty pomocí vytváření programových kódů,
- načítat data z jiných externích zdrojů, případně je exportovat,
- sdílet data jiným uživatelům a další činnosti spojené s databázemi. (Kruczek, 2010)

V podkapitolách této kapitoly jsou popsány základní důležité prvky aplikace Microsoft Access potřebné k vytvoření vlastní databázové aplikace, relační vztahy jsou popsány již ve čtvrté podkapitole kapitoly 2.3.1.

2.5.1. Tabulky

Tabulka představuje dvourozměrnou strukturu skládající se ze sloupců, neboli polí a z řádků, označované též jako záznamy. Pole představuje jeden sloupec tabulky a jedná se tedy o jednu z vlastností popisované věci v tabulce. Záznamem se rozumí jeden řádek tabulky popisující jednu konkrétní věc. Dalším pojmem souvisejícím s tabulkou je položka, která značí jeden konkrétní údaj v daném sloupci a řádku. (Kruczek, 2010)

2.5.1.1. Vytvoření tabulky

Tabulky jsou klíčovým základem relační databáze, a proto je její vytvoření a modifikace významnou činností v aplikaci Microsoft Access. Každá tabulka může mít odlišný účel, obsahově se mohou lišit a jejich použití může probíhat různým způsobem. V Microsoft Accessu lze vytvářet tabulky různými způsoby:

- Jako prázdnou tabulku pomocí tlačítka Tabulka.
- Jako prázdnou tabulku v návrhovém zobrazení pomocí tlačítka Návrh tabulky, což je nejpoužívanějším způsobem s možností definovat jednotlivé sloupce a datové typy.
- Jako tabulku propojenou se službou SharePoint umožňující zpřístupnit tabulky více uživatelům, kteří mohou společně na tabulkách pracovat.
- Pomocí přednastavených šablon, což je vhodné například pro začínající uživatele, případně pokud není čas se zdržovat s definováním tabulky. (Kruczek, 2010)

2.5.1.2. Zobrazení tabulky

V aplikaci Microsoft Access je umožněno tabulky zobrazit různými způsoby:

- **Zobrazení datového listu** – v tomto zobrazení, které je podobnému tabulkám aplikace Excel, je možné sloupce vkládat, formátovat a odstraňovat a také vytvářet nové záznamy, upravovat je, vyhledávat a filtrovat, případně je mazat,
- **Návrhové zobrazení** – toto zobrazení je určeno výhradně pro práci s poli a pro každé pole se vyplňuje jeho název, datový typ a nepovinný údaj popis,
- **Zobrazení kontingenční tabulky** – pomocí kontingenční tabulky je umožněno vytvořit tabulku se záhlavím řádků a sloupců, slouží k analýze dat,
- **Zobrazení kontingenčního grafu** – jedná se o grafické vyjádření kontingenční tabulky.

Nejvíce je používáno návrhové zobrazení a zobrazení datového listu. (Kruczek, 2010), (Písek, 2011)

2.5.2. Datové typy

Volba datového typu je nedílnou a jednou z nejdůležitějších součástí návrhu databáze ovlivňující celé její chování. Jedná se o vlastnost určující chování pole. Existuje několik datových typů obsažených v aplikaci Microsoft Access.

- **Text** – pomocí tohoto datového typu je umožněn zápis libovolných znaků do záznamů tabulky v rozmezí od žádného znaku do 255 znaků, lze zapisovat číslice, písmena a další speciální znaky.
- **Memo** – typ Memo je určen k zápisu dlouhých textových řetězců dlouhých až 65535 znaků, nelze pomocí něj třídit ani filtrovat.
- **Číslo** – slouží k ukládání číselných hodnot a je mu možné nastavit různou velikost, jako je bajt, celé číslo, desetinné číslo, a další.
- **Datum a čas** – je používán k uchovávání datových a časových údajů.
- **Měna** – používá se k ukládání peněžních hodnot a je k němu možné přidat symbol měny.
- **Automatické číslo** – slouží pro automatické generování hodnot primárního klíče a v rámci každé tabulky může být tento typ použit pouze jednou.
- **Ano / Ne** – uchovávají se pomocí něj logické hodnoty značící ano nebo ne, respektive true nebo false.
- **Objekt OLE** – pomocí tohoto typu je umožněno vkládat do databáze komplexní data vytvořena v jiných aplikacích, jedná se například o obrázky, dokumenty Word nebo zvuk.
- **Hypertextový odkaz** – slouží k vložení odkazů na externí soubory buďto pomocí adresy typu URL odkazující na internetovou síť nebo může odkaz směřovat na soubory v lokální síti či uložišti.
- **Příloha** – do pole s tímto datovým typem je umožněno vkládání souborů obdobně jako u příloh u elektronické pošty, příloha je pružnější než objekt OLE a zabírá méně místa.
- **Vypočteno** – slouží k možnosti zadání výpočtového výrazu na základě jiných sloupců tabulky.
- **Průvodce vyhledáváním** – jedná se o průvodce, pomocí kterého se může nastavit vyhledávací pole buď jako seznam hodnot z jiné tabulky nebo jako pevná seznam hodnot. (Kruczek, 2010)

2.5.3. Vlastnosti datových typů

V aplikaci Microsoft Access má každý datový typ určité vlastnosti specifikující zadávání dat do tabulek. Vlastnosti jsou předdefinované, ale lze je dle potřeby měnit. Obecně pro některé vlastnosti platí, že mohou být společné pro různé datové typy.

- **Velikost pole** – slouží k určení velikosti paměti pro dané pole, u textových polí se jedná o maximální počet znaků, které lze do pole zapsat, u datového typu *číslo* typ a přesnost, velikost lze dodatečně měnit, ovšem při zmenšování s rizikem ztráty informace delší, než je nová velikost.
- **Počet desetinných míst** – udává počet zobrazovaných desetinných míst v poli s rozsahem 0 až 15 a souvisí pouze s datovými typy *číslo* a *měna*.
- **Formát** – pomocí této vlastnosti se určuje způsob zobrazení textu v daném poli, lze ho bez následků měnit i dodatečně.
- **Vstupní maska** – slouží k určení, jaké hodnoty je možné do pole zadat a jak se uloží v tabulce a to prostřednictvím zástupných symbolů, jako je například 0 značící povinné číslo, L značící povinné písmeno, a další, vstupní masku je možné zadat ručně nebo pomocí průvodce a pouze pro datové typy *text* a *datum/čas*.
- **Titulek** – jedná se o popisek záhlaví pole a vyplňuje se pouze tehdy, pokud má být odlišný od názvu pole.
- **Výchozí hodnota** – výchozí hodnotou se rozumí automaticky vložený údaj v novém záznamu tabulky.
- **Ověřovací pravidlo** – v této vlastnosti je možné nastavit kritérium, které musí splňovat vložený údaj do pole, jestliže není splněno, objeví se hlášení nastavené ve vlastnosti *Ověřovací text*, například ověřovací pravidlo $<>0$ značí, že zadaná hodnota musí být různá od nuly, a další.
- **Ověřovací text** – jedná se o chybové hlášení objevující se při porušení ověřovacího pravidla.
- **Je nutno zadat** – je-li hodnota této vlastnosti nastavena na Ano, tak je ji nutné vyplnit, jinak nebude možné záznam uložit.
- **Povolit nulovou délku** – pomocí této vlastnosti se povoluje, případně zakazuje vkládat texty nulové délky, je-li zadáno Ano, tak je možné zadat prázdný řetězec, v opačném případě nebude možné záznam uložit.
- **Indexovat** – indexace slouží k urychlení průběhu zpracování dotazů, filtrů a dalších operací se záznamy, lze vybrat ze tří hodnot, a to *Ne*, kdy není vytvořen

žádný index, *Ano (duplicita povolena)*, kdy je vytvořen index a je umožněna existence dvou záznamů se stejnou hodnotou, *Ano (bez duplicity)*, kdy je vytvořen jedinečný index, což neumožňuje vytvořit duplicitní záznamy. (Kruczek, 2010), (Písek, 2011)

2.5.4. Formuláře

Formulář je databázový objekt, který slouží pro efektivní práci s daty v tabulkách a celé databázi a je pomocí něj umožněno data zobrazovat, zadávat a měnit. Lze si jej představit jako okna, pomocí kterých přistupuje uživatel k databázi. Slouží uživatelům k usnadnění orientace při práci s databází a poskytují větší možnost kontroly zadávaných dat. Pomocí každého formuláře se přistupuje právě k jedné tabulce či dotazu, ale je možné do formuláře vložit tzv. podformulář, pomocí kterého je umožněno zobrazit informace z několika tabulek zároveň. (Kruczek, 2010)

Formulář lze vytvořit několika způsoby:

- **Automatický formulář** – tento způsob vytvoření formuláře je nejrychlejším a nejjednodušším, ale je omezen na zdroje dat pouze z jedné tabulky, nemusí však splňovat ani další požadavky a je potřeba ho dále upravovat.
- **Vytvoření pomocí průvodce** – jedná se o snadné vytvoření formuláře prostřednictvím několika kroků v průvodci.
- **Vytvoření ručně v návrhovém zobrazení** – tento způsob je nejnáročnější, avšak lze díky němu vytvořit, případně upravit formulář jak je potřeba.

Aplikace Microsoft Access nabízí více možností jak formulář zobrazit, mezi základní tři možnosti patří formulářové zobrazení, zobrazení rozložení a návrhové zobrazení. Další režimy zobrazení jsou pouze pro speciální typy formulářů.

- **Formulářové zobrazení** – slouží k prohlížení, upravování či vkládání nových dat, zejména standardním uživatelům a nelze v tomto režimu návrh formuláře měnit.
- **Návrhové zobrazení** – v tomto režimu nelze pracovat s konkrétními daty, ale mění se zde návrh formuláře, přidávat nové ovládací prvky, apod..
- **Zobrazení rozložení** – toto zobrazení umožňuje upravovat vlastnosti ovládacích prvků a zároveň se již v jednotlivých polích zobrazují konkrétní data. Jedná se o kombinaci formulářového a návrhového zobrazení.
- **Zobrazení datového listu** – zobrazení formuláře je ve formě datového listu.
- **Zobrazení kontingenční tabulky** – je zobrazen jako kontingenční tabulka.

- **Zobrazení kontingenčního grafu** – pomocí tohoto zobrazení si lze prohlížet data v kontingenčním grafu. (Kruczek, 2010), (Písek, 2011)

V rámci formulářů lze vkládat různé ovládací prvky dělicí se na *vázané*, které slouží k zobrazování údajů mající nějaký vztah k datům v databázi a *nevázané*, které zobrazují data bez vztahu s databází. (Písek, 2011)

2.5.5. Filtry

Aby bylo možné pracovat s určitými daty z databáze, je potřeba tyto data nějakým způsobem z databáze vybrat. Nejrychlejším a také nejjednodušším způsobem výběru dat jsou tzv. filtry. Umožňují vybrání a zobrazení části dat odpovídající zadaným kritériím. Ukládají se jako vlastnost tabulky. V rámci aplikace Access jsou různé typy filtrů.

- **Filtr podle výběru** – jedná se o nejrychlejší vyhledávání záznamů se společnými rysy, například záznamy se shodnou částí textu.
- **Filtr mimo výběr** – podobá se filtru podle výběru s rozdílem, že jsou vybrány všechny záznamy neodpovídající kritériu.
- **Filtr podle formuláře** – je vhodné ho použít, když je potřeba vybírat hodnoty ze seznamu, případně když je nutné použít složitou podmínku.
- **Rozšířený filtr či řazení** – slouží k vyhledávání dat podle složitějších kritérií a v definici filtru je možné nastavit způsob řazení dat. (Písek, 2011)

2.5.6. Dotazy

Dotaz je účinný nástroj pro práci s databází jak pro uživatele, tak pro správce a je určen k veškeré manipulaci s daty v databázi. Je vhodné je používat, když je potřeba vybírat data podle složitějších kritérií. Dotazy jsou na rozdíl od filtru samostatným objektem. Charakteristické pro dotazy je:

- Dotaz je možné uložit pro pozdější použití.
- Pomocí dotazu je umožněno vybírat data libovolně z celé databáze.
- Jeho výsledkem je samostatné okno s datovým listem obsahující vybrané záznamy a lze jej použít jako vstupní data pro tvorbu dalšího dotazu.
- Prostřednictvím dotazu lze vytvořit souhrny, jako jsou například mezisoučty.
- Aplikace Microsoft Access vytvořené dotazy v uživatelském rozhraní převádí do jazyka SQL automaticky, ale je také možné tvořit dotazy přímo v jazyce SQL. (Kruczek, 2010), (Písek, 2011)

Dotazy je umožněno vytvářet buď pomocí průvodce, pomocí kterého lze dotaz vytvořit rychle a efektivně, nebo manuálně, kdy se musí všechny parametry nastavit ručně. Vhodné je vytvořit dotaz pomocí průvodce a posléze ho upravovat ručně. (Kruczek, 2010)

V aplikaci Microsoft Access je možné pro různé úkoly použít různé typy dotazů.

- **Výběrový dotaz** – jedná se o nejčastěji používaný dotaz a často se od něj odvozují ostatní typy dotazů a slouží k výběru dat, může být podrobný, což znamená, že zobrazí veškerá vybraná pole a záznamy s nimi související, nebo souhrnný, pomocí kterého lze zpracovat průměry, mezisoučty a počty záznamů dané seskupení, výsledkem tohoto dotazu může být pole z více tabulek.
- **Křížový dotaz** – výsledkem tohoto speciálního typu dotazu je křížový tabulkový list obsahující vzájemnou závislost dvou různých údajů, tento dotaz může sloužit jako základ pro tvorbu grafů.
- **Akční dotazy** – pomocí těchto dotazů se upravují data v databázi, lze upravovat najednou velké množství těchto dat, v Accessu je možné použít čtyři typy těchto dotazů.
 - **Odstraňovací dotaz** – slouží k odstranění části záznamů z tabulky, které byly dle zadaných kritérií vybrány.
 - **Aktualizační dotaz** – slouží k vykonání změny určitých již existujících dat v databázi na základě zadaných kritérií.
 - **Přidávací dotaz** – pomocí tohoto dotazu je umožněno přidávat záznamy výsledku dotazu do tabulky.
 - **Vytvářecí dotaz** – lze pomocí něj vytvořit zcela novou tabulku, která bude obsahovat pole a záznamy z výsledku dotazu.
- **Dotazy SQL** – jedná se o zvláštní typ dotazů vytvářené standardním dotazovacím jazykem, jejich tvorba je odlišná a připomíná spíše programování, existují tři kategorie těchto dotazů.
 - **Sjednocovací dotaz** – slouží k tvorbě základních dotazů pro výběr záznamů a spojování tabulek.
 - **Předávací dotaz** – slouží pouze k předání dotazu na databázový stroj a zapisují se v jeho syntaxi.
 - **Definiční dotaz** – slouží k práci s tabulkami, jako akční dotazy, jen s více možnostmi. (Kruczek, 2010), (Písek, 2011)

2.5.7. Sestavy

Sestava je dalším efektivním nástrojem aplikace Access, slouží k prezentování dat z databáze i v tištěné podobě. Sestavě se může navrhovat její vzhled, rozložení, jaké data bude zobrazovat, ale žádným způsobem tyto data neovlivňuje. V rámci aplikace Access lze sestavu zobrazit obdobně jako formulář, a to jako zobrazení sestavy, náhled, zobrazení rozložení a návrhové zobrazení. Vytvořit sestavu lze taktéž obdobnými možnostmi jako tomu je u formulářů, a těmi možnostmi jsou základní sestava, průvodce sestavou a úprava sestavy v návrhovém zobrazení. (Kruczek, 2010), (Písek, 2011)

2.5.8. Makra

Makra jsou dalším objektem aplikace Access sloužící k automatizaci práce s databází. Jedná se o posloupnost po sobě jdoucích akcí v daném pořadí vykonávajících se za určitých podmínek. Makra lze vytvořit relativně snadno a další výhodou je, že v české verzi Microsoft Access je lze tvořit v češtině, ovšem akce dostupné díky makra jsou omezené. Pomocí programových modulů jazyka VBA, které lze přirovnat k makrům, je umožněna takřka neomezená práce s databází a tudíž lze tvořit složitější úkony, než pomocí makra, ovšem je nutné znát programovací jazyk VBA. (Kruczek, 2010)

2.5.9. Visual Basic for Application (VBA)

VBA neboli Visual Basic for Application je procedurální strukturovaný programovací jazyk vyvinutý společností Microsoft a jedná se o intuitivnější, robustnější a výkonnější verzi jazyka Visual Basic, ze kterého je odvozen a je více objektově-orientovaný než starší jazyky pro tvorbu maker. Pomocí jazyka VBA je umožněno psát makra pro všechny aplikace systému Office a zejména je umožněno rozšířit funkcionalitu aplikace Microsoft Access. (Shepherd, 2012)

Zdrojový kód jazyka VBA je v aplikaci Access možné psát v přidruženém programu nazvaném Visual Basic Editor do tzv. modulů.

- **Standardní moduly** – kód napsaný ve standardních modulech je přístupný v každé části aplikace a zapisují se zde univerzální funkce nebo procedury,
- **Moduly třídy** – většinou to jsou moduly formulářů a sestav. (Shepherd, 2012), (Novák, 2014)

3. Analýza současného stavu ve firmě

V této kapitole je představena firma, pro kterou je databázová aplikace tvořena a popis současné situace, respektive způsob, jakým jsou data ve firmě aktuálně uchovávána a zpracovávána. Nejdůležitější součástí kapitoly je poslední podkapitola, ve které jsou popsány požadavky zadavatele, které by měla nově vytvářená databázová aplikace splňovat.

3.1. O společnosti

Firma RadEX Primar s.r.o. je specializovaná zkušební laboratoř pověřená Státním úřadem pro jadernou bezpečnost. Předmětem zkoušek prováděných zkušební laboratoří jsou diagnostické zdravotnické rentgenové přístroje a rentgenová pracoviště jako celky. Firma provádí kontroly rentgenových přístrojů na úrovni přejímacích zkoušek, zkoušek dlouhodobé stability a zkoušek provozní stálosti, dále kontroly rentgenových pracovišť, jako je monitorování míst významných z hlediska ochrany před zářením, zpracovává kompletní dokumentace a vytváří a zavádí programy zabezpečování jakosti, provádí kvalifikaci osob, jako je příprava ke zkoušce zvláštní odborné způsobilosti, školení, apod. a také poradenskou činnosti.

3.2. Současná situace

Zkušební laboratoř pracuje v terénu s více pracovními skupinami. Každá z těchto skupin je vybavena vlastní komplexní sadou měřicího a testovacího vybavení, pomocí kterého provádí zkoušky zařízení u klientů. Pracovní skupiny má za úkol delegovat jeden pracovník, který na základě nashromážděných informací domlouvá a koordinuje provádění zkoušek u jednotlivých klientů a určuje optimální trasu. Primární informací, podle které se domlouvá provádění zkoušek, je datum posledního provádění této zkoušky, respektive takzvaného měření RTG zařízení. Zkoušky rentgenových zařízení se musí provádět v pravidelných intervalech daných atomovým zákonem a jeho prováděcími předpisy.

Aktuálně jsou informace o zákaznících uchovávány pomocí programu Microsoft Access verze 2000 v databázi s příponou mdb. Tato původní databáze byla vytvořena před více než 15 lety zaměstnancem firmy, který měl pouze základní znalosti v oblasti tvorby databázových aplikací v programu Microsoft Access, obsahuje pouze základní informace o zákaznících a jejich rentgenových přístrojích a zdaleka nevyhovuje současným požadavkům firmy. Mnohé informace si firma uchovává také pomocí programů Microsoft Excel a Microsoft Word. Některé potřebné informace jsou pro zaměstnance těžko

dohledatelné nebo nepřehledné, případně některé úkony jsou zbytečně náročné. Proto je potřeba vytvořit novou databázovou aplikaci, která by měla mít daleko větší spektrum uchovávaných informací a také více možností, jak s jednotlivými informacemi nakládat a mnohé věci zautomatizovat. Původní databáze při tvorbě nové databázové aplikace nebyla firmou poskytnuta z důvodu ochrany osobních údajů zákazníků.

3.3. Požadavky na aplikaci

Jelikož byla původní databáze firmy vytvořena značně neprofesionálně a nebyla poskytnuta možnost do ní nahlédnout kvůli ochraně osobních údajů zákazníků, byl vznesen požadavek, aby byla vytvořena úplně nová databázová aplikace.

Prioritními informacemi, které by měly být v databázové aplikaci uchovávány, jsou podrobné informace o jednotlivých zákaznících a jejich rentgenových přístrojích. Mezi zákazníky firmy patří jak fyzické osoby (zubní lékaři, veterinární lékaři), tak právnické osoby jako zdravotnická zařízení (nemocnice, polikliniky, s.r.o., apod.), a o každém z nich by měly být uchovávány vyjma klasických informací jako je jméno, adresa apod., i odborné informace spojené s problematikou, jíž se firma zabývá. Mělo by být umožněno u každého zákazníka vkládat několik poznámek s aktuálním časem vložení.

U každého zákazníka by se měly zobrazovat všechny jeho rentgenové záříče, kterých může mít i více a konkrétní informace o nich. Jednotlivá RTG zařízení by měla mít k dispozici historii jejich měření, respektive provádění zkoušek, kterou bude na základě měření z minulých období zaměstnanec vkládat. Tyto všechny informace by se měly zobrazovat v jednom formuláři, neboť jsou klíčové.

Součástí databázové aplikace by měla být i tabulka se zaměstnanci, kteří provádí měření u zákazníků, případně vykonávají zákazníkovi tzv. dohled. U rentgenových zařízení by mělo být uvedeno, který pracovník provedl poslední měření a u zákazníka, kterému dělá firma dohled, by mělo být uvedeno, kdo ze zaměstnanců je touto dohlízející osobou.

Veškeré záznamy v databázové aplikaci by mělo být možné upravovat, odstraňovat a přidávat nové, a to i člověkem se základní uživatelskou znalostí informačních technologií. Změny v databázové aplikaci bude provádět pouze jeden zaměstnanec, ostatní zaměstnanci budou mít po domluvě s tímto pracovníkem možnost do ní pouze nahlížet.

Dále je potřeba, aby si zaměstnanec jednoduše zobrazil přehled zákazníků měřených v určitém měsíci, aby na základě toho mohl domlouvat se zákazníky další termín měření jejich rentgenového zařízení. Jak již bylo zmíněno dříve v této bakalářské práci, intervaly

měření jsou dány atomovým zákonem a měření probíhá zpravidla ve stejném měsíci, v jakém bylo provedeno v minulém období. V databázové aplikaci by dále mělo být umožněno zobrazit souhrn zákazníků, kteří byli v určitém měsíci proškoleni, souhrn zákazníků z určitého města, u jednotlivých zaměstnanců zobrazit souhrn zákazníků, kterým dělá daný zaměstnanec dohled, komu měřil RTG zařízení za určitý měsíc a také celkový souhrn zaměstnancova měření.

Jako bonusový požadavek, který nemusí být splněn, ale výrazně by urychlil práci, uvedl zadavatel možnost exportování konkrétních informací o zákazníkovi z databáze do sešitu souboru Excel, který má dán pevnou strukturu. Jedná se o průvodní list zákazníka, který slouží jako podklad pro další interní úkony ve firmě. Doposud jsou informace o zákazníkovi vypisovány do průvodního listu ručně, což zbytečně zaměstnance časově vytěžuje.

Celá požadovaná databázová aplikace by měla práci usnadnit a zpřehlednit a její ovládání by mělo být intuitivní a snadno osvojitelné.

Firma disponuje programovým vybavením Microsoft Office 2010 a z toho důvodu je databázová aplikace tvořena právě v programu Microsoft Access verze 2010.

4. Návrh a realizace databázové aplikace

Tato kapitola je stěžejní kapitolou celé bakalářské práce a je v ní popsán návrh a následná praktická tvorba databázové aplikace. V první části je návrh konceptuálního modelu, který je následně převeden do logického modelu. Oba tyto modely jsou vytvořeny pomocí entitně-relačního diagramu.

Dalším krokem je převod logického návrhu do fyzického modelu, konkrétně do prostředí programu Microsoft Access verze 2010. Klíčové prvky tvořené právě v programu Microsoft Access 2010 jsou popsány v další části kapitoly, konkrétně se jedná o popis vytvořených tabulek, které jsou podrobněji popsány, a je ujasněn jejich smysl. Do tabulek databáze jsou pro účely bakalářské práce vložena pouze zkušební data sloužící k testování funkčnosti aplikace.

Následují části věnované konkrétní funkčnosti tvořené databázové aplikace. Jednotlivě jsou zde popsány veškeré formuláře, kterým je jejich funkčnost zajištěna prostřednictvím programového kódu jazyka VBA napsaného ve vývojovém prostředí přidruženého programu VBE. Všechny formuláře jsou jednoduše graficky upraveny, aby bylo dosaženo přívětivého uživatelského prostředí. V této části kapitoly jsou i popsány vytvořené dotazy, ze kterých jsou následně tvořeny sestavy.

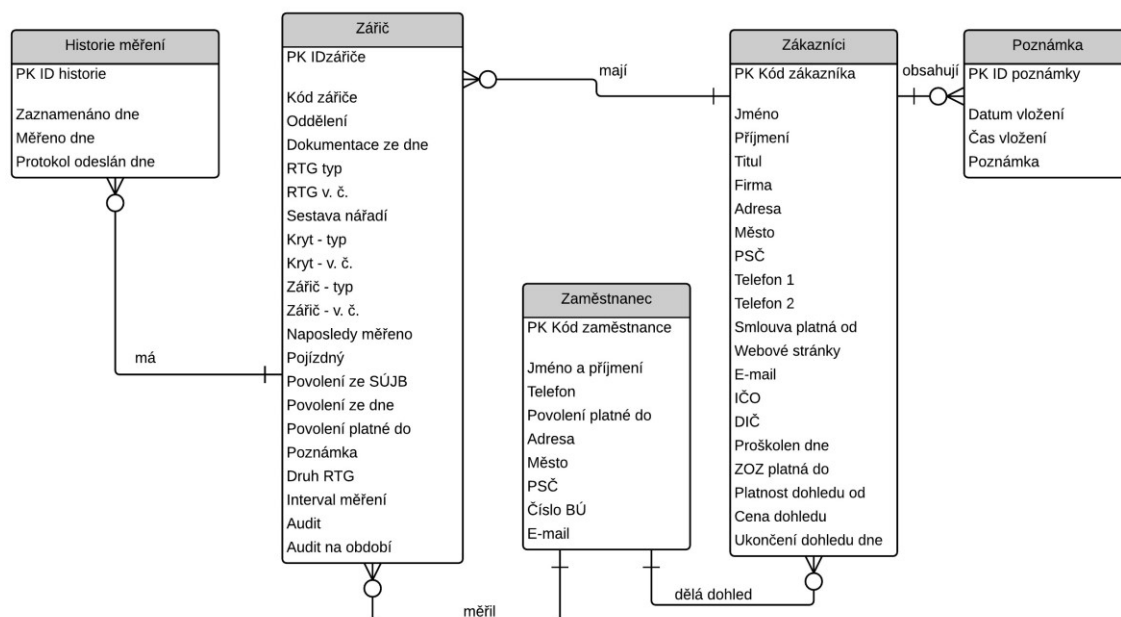
Při praktické tvorbě celé databázové aplikace je vycházeno z teoretické části, která byla popsána ve 2. kapitole této bakalářské práce.

V závěru této kapitoly je stručně popsáno, jakým způsobem bude výsledná databázová aplikace implementována do firmy.

4.1. Konceptuální návrh

Návrh konceptuálního modelu byl tvořen na základě požadavků vznesených jednatelem firmy. Byly určeny základní entity mající pro firmu určitý význam, a o kterých je potřeba shromažďovat informace a vztahy mezi těmito entitami včetně jejich typu. Entity jsou dále popsány svými atributy.

Na obrázku 4.1 lze vidět konceptuální model k této bakalářské práci. Jsou v něm obsaženy všechny důležité entity, které jsou vždy nahoře pojmenovány, a jejich atributy. Každá z těchto entit má umístěn pod svým názvem určený primární klíč (značeno zkratkou PK). Z obrázku jsou patrné i vztahy mezi entitami, které značí jejich vzájemné propojení. Kardinalita všech vztahů je 1:N.



Obrázek 4.1: Návrh konceptuálního modelu, zdroj: vlastní

4.1.1. Přehled entit

V této podkapitole je stručně popsáno všech pět entit a atributy v daných entitách. Jedná se o atributy, které byly na základě zadavatelova požadavku nejdůležitější.

Zákazníci

Jedná se o entitu reprezentující zákazníka zkušební laboratoře. Zákazníky zkušební laboratoře jsou fyzické i právnické osoby, kterým firma provádí nějakou službu z jejich portfolia.

Jako primární klíč byl zvolen kód zákazníka, který firma interně přiděluje každému zákazníkovi jako jedinečnou identifikaci. Entita Zákazníci má vazby na entitu Záříč a entitu Poznámka typu 1:N, což znamená, že jeden zákazník může mít více záříčů a je umožněno k němu psát více nezávislých poznámek. Tato entita je ve vztahu i s entitou Zaměstnanec 1:N, s tím rozdílem, že nyní se nachází na straně N, a to z toho důvodu, že několika zákazníkům může dělat tzv. dohled jeden zaměstnanec.

Atributy entity Zákazníci a jejich popis:

- kód zákazníka – interně přidělený jedinečný identifikátor zákazníka, jedná se o primární klíč,
- jméno – značí jméno fyzické osoby, případně kontaktní osoby (pokud se jedná o právnickou osobu),
- příjmení – příjmení fyzické osoby, případně kontaktní osoby,

- titul – představuje dosažený titul zákazníka,
- firma – název právnické osoby,
- adresa – značí ulici a číslo popisné, kde zákazník vykonává činnost,
- město – představuje město, kde zákazník vykonává činnost,
- PSČ – poštovní směrovací číslo,
- telefon 1 – představuje telefonní kontakt na zákazníka,
- telefon 2 – druhý telefonní kontakt, v případě, že má více telefonních čísel,
- smlouva platná od – od jakého data je platná smlouva uzavřena mezi zákazníkem a zkušební laboratoří,
- webové stránky – představuje internetové stránky zákazníka,
- e-mail – e-mailová adresa zákazníka,
- IČO – identifikační číslo osoby, které má zákazník ze zákona přiděleno,
- DIČ – daňové identifikační číslo, přiděleno firmě dle zákonných podmínek,
- proškolen dne – datum, kdy byl zákazník proškolen,
- ZOZ platná do – datum, do kdy je platná zkouška odborné způsobilosti,
- platnost dohledu od – datum, od kdy je u zákazníka vykonáván dohled,
- cena dohledu – představuje smluvní cenu za dohled,
- ukončení dohledu dne – datum, ke kterému dni byl dohled ukončen.

Záříč

Další entita slouží k reprezentaci rentgenových záříčů, které vlastní zákazníci. Jedná se o diagnostické zdravotnické rentgenové přístroje vykazující ionizující záření. Tyto přístroje musí být pravidelně kontrolovány a je na nich prováděno tzv. měření, respektive zkoušky, jejichž provádění je hlavní činností zkušební laboratoře, pro kterou je databázová aplikace tvořena.

Primárním klíčem této entity je automaticky přidělované číslo (ID záříče). Entita Záříč má vazbu 1:N na entitu Historie měření, z čehož vyplývá, že jeden záříč může mít několik záznamů v historii jeho měření. Tato entita je dále ve vztahu typu 1:N s entitami Zákazníci a Zaměstnanec, ale tentokrát na straně N, což znamená, že více záříčů může být vlastněno jedním zákazníkem, respektive několik záříčů může být měřeno jedním zaměstnancem zkušební laboratoře.

Atributy entity Záříč a jejich popis:

- ID záříče – primární klíč přidělovaný automaticky,
- kód záříče – kód pro interní identifikaci ve firmě, složený z kódu zákazníka, kterému záříč patří a dále z čísla značícího, o který záříč v pořadí se u daného zákazníka jedná,
- oddělení – na kterém oddělení se záříč nachází (v případě, že se jedná například o nemocnici),
- dokumentace ze dne – datum, ze kterého dne byla vypracována k záříči dokumentace,
- RTG typ – o jaký typ rentgenu se jedná,
- RTG v. č. – výrobní číslo rentgenu,
- sestava nářadí – značí, jakými diagnostickými přístroji se provádí měření záříče,
- kryt – typ – představuje, jaký kryt daný záříč má,
- kryt – v. č. – výrobní číslo krytu záříče,
- záříč – typ – jedná se o rentgenku záříče,
- záříč – v. č. – výrobní číslo rentgenky záříče,
- naposledy měřeno – datum, kdy bylo naposledy provedeno měření RTG záříče,
- pojízdný – volba, zda se jedná o pojízdný RTG přístroj,
- povolení ze SÚJB – značí číslo povolení ze Státního úřadu pro jadernou bezpečnost, povolení je dokument potvrzující, že daná osoba (například zubní lékař) může s daným rentgenovým zařízením nakládat,
- povolení ze dne – datum, kdy bylo povolení úřadem vydáno,
- povolení platné do – do kdy je povolení platné, může se jednat o datum, případně může být platné neomezeně,
- poznámka – jedná se o poznámku k danému záříči,
- druh RTG – o jaký druh zdroje ionizujícího zařízení se jedná, příkladem může být významný zdroj, jednoduchý, apod.
- interval měření – značí, jak často se daný rentgenový záříč musí měřit,
- audit – volba, zda má zákazník k určitému záříči audit, má-li audit, tak nemusí nechávat v daném období záříč měřit,
- audit na období – značí, na jaké období je audit.

Zaměstnanec

Entita Zaměstnanec reprezentuje zaměstnance firmy, kteří provádí měření rentgenových zařízení, jedná se o tzv. technické pracovníky, administrativní a podobné zaměstnance není třeba v databázi uvádět.

Primárním klíčem je kód zaměstnance, který je složen z iniciály zaměstnance a jednoho čísla (pro případ, že bude více zaměstnanců se stejnými iniciálami). Entita Zaměstnanec je ve vztahu typu 1:N s entitami Zákazníci a Zářič, což znamená, že jeden zaměstnanec může měřit více zářičů, respektive může dělat dohled více zákazníkům.

Atributy entity Zaměstnanec a jejich popis:

- Kód zaměstnance – primární klíč složený z iniciály zaměstnance a čísla,
- jméno a příjmení – značí jméno a příjmení zaměstnance,
- telefon – představuje telefonní kontakt na zaměstnance,
- povolení platné do – datum, do kdy má zaměstnanec platné povolení opravňující ho provádět měření rentgenových zářičů,
- adresa – značí ulici a číslo popisné bydliště zaměstnance,
- město – představuje město jeho bydliště,
- PSČ – poštovní směrovací číslo bydliště,
- číslo BÚ – číslo bankovního účtu zaměstnance,
- e-mail – e-mailová adresa zaměstnance.

Poznámka

Tato entita reprezentuje jednotlivé poznámky, které lze k zákazníkům psát. Primárním klíčem je automaticky přidělované číslo (ID poznámky). Entita Poznámka má vazbu pouze k entitě Zákazníci, a to typu 1:N, kdy entita Poznámka se nachází na straně N, což umožňuje několik poznámek psát k jednomu zákazníkovi.

Atributy entity Poznámka a jejich popis:

- ID poznámky – primární klíč přidělovaný automaticky,
- datum vložení – představuje datum, kdy byla poznámka k zákazníkovi vložena,
- čas vložení – představuje čas, kdy byla poznámka vložena,
- poznámka – značí samotný text poznámky.

Historie měření

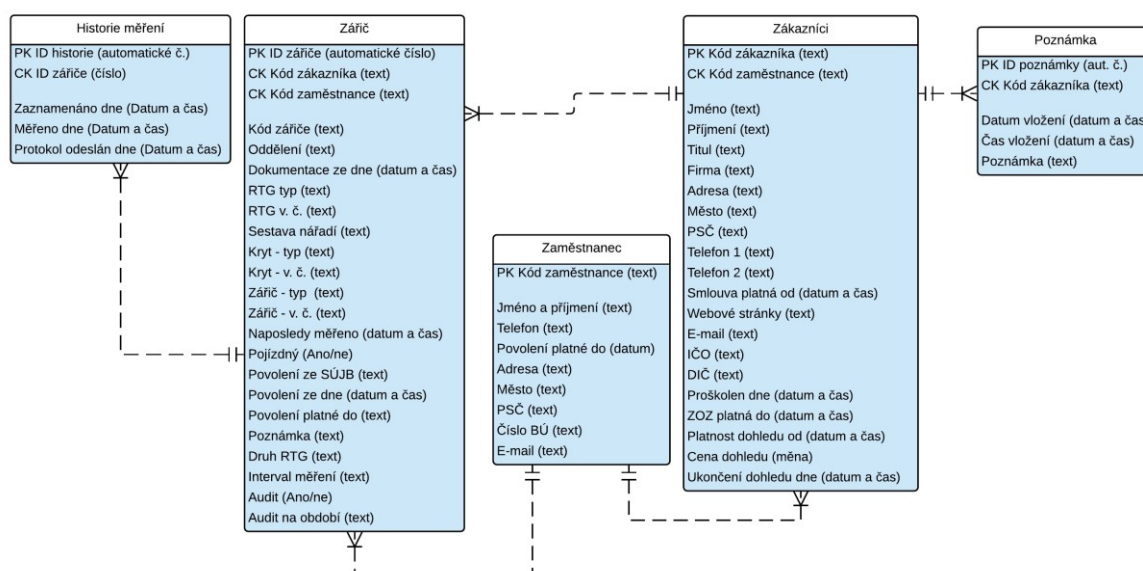
Poslední entita představuje jednotlivé záznamy z historie měření daného rentgenového zařízení. Primárním klíčem je automaticky přidělované číslo (ID historie). Tato entita je ve vztahu 1:N na straně N s entitou Zářič, jelikož několik záznamů v historii může být u jednoho určitého zářiče.

Atributy entity Historie měření a jejich popis:

- ID historie – automaticky přidělovaný primární klíč,
- zaznamenáno dne – představuje datum, kdy byl záznam o historii měření přidán,
- měřeno dne – datum, kdy bylo provedeno měření,
- protokol odeslán dne – datum, kdy byl zákazníkovi odeslán protokol o měření jeho zářiče.

4.2. Logický návrh

Návrh logického modelu byl tvořen na základě výstupu z modelu konceptuálního. Stejně jako konceptuální model je logický model vyobrazen ve formě entitně-relačního diagramu. Entity byly převedeny do konkrétních tabulek a zůstal zachován jejich počet. Z atributů entit se nyní staly sloupce tabulek. Obrázek 4.2 představuje konkrétní logický model k této bakalářské práci. Jednotlivé tabulky mají kromě primárních klíčů již přidělené i cizí klíče tabulek s nimi propojených a sloupcům byly přiděleny příslušné datové typy.



Obrázek 4.2: Návrh logického modelu, zdroj: vlastní

4.3. Aplikace Microsoft Access

Posledním krokem datového modelování je vytvoření fyzického modelu, který vychází z výstupů logického modelu. Tato kapitola je zaměřena již na praktickou tvorbu databázové aplikace v programu Microsoft Access 2010. Tabulky z logického modelu byly převedeny již do konkrétních tabulek v programu Microsoft Access 2010 a byly propojeny vzájemnými vazbami.

Budou zde popsány jednotlivé tabulky databáze, vztahy mezi nimi a vytvořené aplikační prostředí, pomocí kterého je zabezpečena práce s tabulkami. Konkrétně se jedná o formuláře, dotazy a sestavy tvořené na základě těchto dotazů.

4.3.1. Tabulky

Tabulky jsou základním prvkem relační databáze, a proto je jejich navržení velmi důležité. Tato podkapitola je právě na ně zaměřena a jsou v ní popsány všechny tabulky převedené z logického modelu do prostředí programu Microsoft Access. Na základě těchto tabulek je posléze vytvořena celá databázová aplikace.

Tabulka *tblZakaznici*

V této tabulce jsou uchovávány informace související se zákazníky zkušební laboratoře. Mezi tyto zákazníky patří zubní lékaři, veterináři, nemocnice, polikliniky, a další zdravotnická zařízení, případně fyzické osoby, které vlastní diagnostický zdravotnický rentgenový přístroj, a kterým firma provádí měření těchto přístrojů, případně zajišťuje jinou službu.

Pole *Kod* je primárním klíčem tabulky a jedná se o interně přidělovaný třímístný kód sloužící k identifikaci zákazníka ve firmě. Toto pole je jako jediné indexované bez duplicity a je jedinou povinnou položkou. Ostatní údaje jsou nepovinné, jelikož ne vždy při vkládání nového zákazníka do databáze jsou známy veškeré údaje a jsou doplňovány dodatečně. U všech textových polí je přiřazena jejich maximální délka. Polím v této tabulce s datovými typy Datum a čas je nastaven formát na Datum (krátké). U polí *PSC*, *Tel1* a *Tel2* jsou nastaveny vstupní masky určující, jaké hodnoty se mohou do pole zapsat. Pole *Kod_Zam* je cizím klíčem, pomocí kterého je tabulka *tblZakaznici* propojena s tabulkou *tblZamestnanci*, což je nutné, aby bylo možné u zákazníka určit, který pracovník mu dělá tzv. dohled. V tabulce 4.1 jsou vypsány veškerá pole, jejich datový typ, velikost, vstupní maska, zda je toto pole nutno zadat a indexace tohoto pole.

Název pole	Datový typ	Velikost pole	Nutno zadat	Vstupní maska	Indexovat
Kod	Text	3	Ano	-	Ano (bez duplicity)
Jmeno	Text	40	Ne	-	Ne
Prijmeni	Text	40	Ne	-	Ne
Titul	Text	15	Ne	-	Ne
Firma	Text	100	Ne	-	Ne
Adresa	Text	100	Ne	-	Ne
Mesto	Text	50	Ne	-	Ne
PSC	Text	5	Ne	000\ 00	Ne
Tel1	Text	9	Ne	000\ 000\ 000	Ne
Tel2	Text	9	Ne	000\ 000\ 000	Ne
Smlouva_z	Datum a čas	-	Ne	-	Ne
Web	Text	60	Ne	-	Ne
Email	Text	50	Ne	-	Ne
ICO	Text	10	Ne	-	Ne
DIC	Text	15	Ne	-	Ne
Proskol_DO	Datum a čas	-	Ne	-	Ne
ZOZ_do	Datum a čas	-	Ne	-	Ne
Platnost_do	Datum a čas	-	Ne	-	Ne
Cena_DO	Měna	-	Ne	-	Ne
Ukoncen	Datum a čas	-	Ne	-	Ne
Kod_Zam	Text	3	Ne	-	Ne

Tabulka 4.1: Návrh tabulky tblZakaznici, zdroj: vlastní

Tabulka *tblZarice*

Tabulka slouží k uchovávání informací o zákaznickovu rentgenovém zářiči.

Jako primární klíč této tabulky je pole *ID_Zarice*, které je přidělováno automaticky při přidávání nového záznamu do tabulky. Stejně jako u tabulky *tblZakaznici*, je i zde nutné zadat pouze jedno pole, a to *Kod_Zarice* složené z kódu zákazníka a číslem značící pořadové číslo zářiče u onoho zákazníka. Ostatní údaje jsou nepovinné ze stejného důvodu jako tomu je u *tblZakaznici*. Polím s datovým typem Datum a čas je přiřazen formát na Datum (krátké). U pole *Druh_RTG* je nastaven jako zobrazovací prvek Pole se seznamem a jako typ zdroje řádku je Seznam hodnot, zdroj řádku je nastaven na výběr hodnot "Jednoduchý zdroj", "Významný zdroj", "Veterinární RTG", díky čemuž je umožněno, že při vkládání hodnot se vybírá z oněch nabízených variant. Pole *Interval_mereni* je nastaven obdobně jako pole *Druh_RTG*, s tím rozdílem, že zdroj řádku je nastaven na hodnoty "1 rok", "2 roky", "3 roky". Pole *Audit_z* má nastavenou vstupní masku, pomocí které se má zapisovat měsíc a rok, na který je audit vyřizován. Má-li zákazník na konkrétní zářič vyřízený audit, jedno období ho nemusí nechávat měřit. Pole *Kod* je cizím klíčem z tabulky *tblZakaznici*, aby bylo možné zákazníkům přiřadit jejich zářiče. Pole *Kod_Zam* je cizím klíčem z tabulky *tblZamestnanci*, což umožňuje u zářiče určit, který zaměstnanec provedl poslední měření. V tabulce 4.2 jsou vypsány veškerá pole této tabulky a podrobnosti k těmto polím.

Název pole	Datový typ	Velikost pole	Nutno zadat	Vstupní maska	Indexovat
ID_Zarice	Automatické číslo	Dlouhé celé číslo	-	-	Ano (bez duplicity)
Kod	Text	3	Ne	-	Ne
Kod_Zarice	Text	6	Ano	LLL\C00	Ano (bez duplicity)
Oddeleni	Text	60	Ne	-	Ne
Dokum_Vyprac	Datum a čas	-	Ne	-	Ne
RTG_Typ	Text	55	Ne	-	Ne

RTG_Cislo	Text	35	Ne	-	Ne
Sestava	Text	55	Ne	-	Ne
Kryt_Typ	Text	55	Ne	-	Ne
Kryt_Cislo	Text	35	Ne	-	Ne
Zaric_Typ	Text	55	Ne	-	Ne
Zaric_Cislo	Text	35	Ne	-	Ne
Mereno	Datum a čas	-	Ne	-	Ne
Pojizdny	Ano/ne	-	Ne	-	Ne
Povoleni	Text	30	Ne	-	Ne
Ze_Dne	Datum a čas	-	Ne	-	Ne
Plati_do	Text	35	Ne	-	Ne
Poznamka	Text	255	Ne	-	Ne
Druh_RTG	Text	35	Ne	-	Ne
Interval_mereni	Text	15	Ne	-	Ne
Kod_Zam	Text	3	Ne	-	Ne
Audit	Ano/ne	-	Ne	-	Ne
Audit_z	Text	6	Ne	00V0000	Ne

Tabulka 4.2: Návrh tabulky *tblZarice*, zdroj: vlastní

Tabulka *tblZamestnanci*

Tabulka slouží k zaznamenávání základních informací o technických pracovnících zkušební laboratoře, kteří provádějí měření rentgenových zařízení a dělají zákazníkům dohled.

Primárním klíčem je pole *Kod_Zam*, které je tvořeno prvními písmeny ze jména a příjmení zaměstnance a z jednoho čísla, které by mělo sloužit k odlišení případných více zaměstnanců se stejnými iniciálami. Pole *Povol_Platne* má formát nastaveno na Datum (krátké). V tabulce 4.3 jsou vypsány veškerá pole, jejich datový typ, velikost a další.

Název pole	Datový typ	Velikost pole	Nutno zadat	Vstupní maska	Indexovat
Kod_Zam	Text	3	Ano	-	Ano (bez duplicity)
Jmeno	Text	60	Ano	-	Ne
Tel	Text	9	Ne	000\ 000\ 000	Ne
Povol_Platne	Datum a čas	-	Ne	-	Ne
Adresa	Text	100	Ne	-	Ne
Město	Text	50	Ne	-	Ne
PSC	Text	5	Ne	000\ 00	Ne
CBU	Text	20	Ne	000000\ - 00000000000\ 0000	Ne
Email	Text	50	Ne	-	Ne

Tabulka 4.3: Návrh tabulky *tblZamestnanci*, zdroj: vlastní

Tabulka *tblPoznamky*

Tato tabulka slouží k ukládání jednotlivých poznámek u zákazníků. Tabulka je vytvořena z důvodu požadavku zadavatele, aby byla možnost u zákazníka psát více na sobě nezávislých poznámek.

Primárním klíčem je pole *ID*, které je automaticky přidělovaným číslem. Pole *Datum* je ve formátu Datum (krátké) a udává datum přidání poznámky. Pole *Cas* je ve formátu Čas (dlouhý) a udává přesný čas přidání poznámky. Pole *Kod* je cizím klíčem z tabulky *tblZakaznici*, což umožňuje jednomu zákazníkovi přiřazovat více poznámek. V tabulce 4.4 jsou uvedeny a několika vlastnostmi popsány veškerá pole této tabulky.

Název pole	Datový typ	Velikost pole	Nutno zadat	Vstupní maska	Indexovat
ID	Automatické číslo	Dlouhé celé číslo	-	-	Ano (bez duplicity)

Kod	Text	3	Ano	-	Ne
Datum	Datum a čas	-	Ano	-	Ne
Cas	Datum a čas	-	Ano	-	Ne
Poznamky	Text	255	Ano	-	Ne

Tabulka 4.4: Návrh tabulky *tblPoznamky*, zdroj: vlastní

Tabulka *tblHistorie*

Poslední tabulka slouží k uchování záznamů o historii měření jednotlivých rentgenových zářičů.

Jako primární klíč je u této tabulky nastaveno pole *ID* s automaticky přidělovaným číslem. Veškeré datové typy Datum a čas v této tabulce jsou ve formátu Datum (krátké). Pole *ID_Zarice* je cizím klíčem, pomocí kterého je tabulka propojena s tabulkou *tblZarice*, aby bylo umožněno jednomu zářiči přiřadit více záznamů historie. V tabulce 4.5 jsou obsaženy veškerá pole této tabulky a jejich vlastnosti.

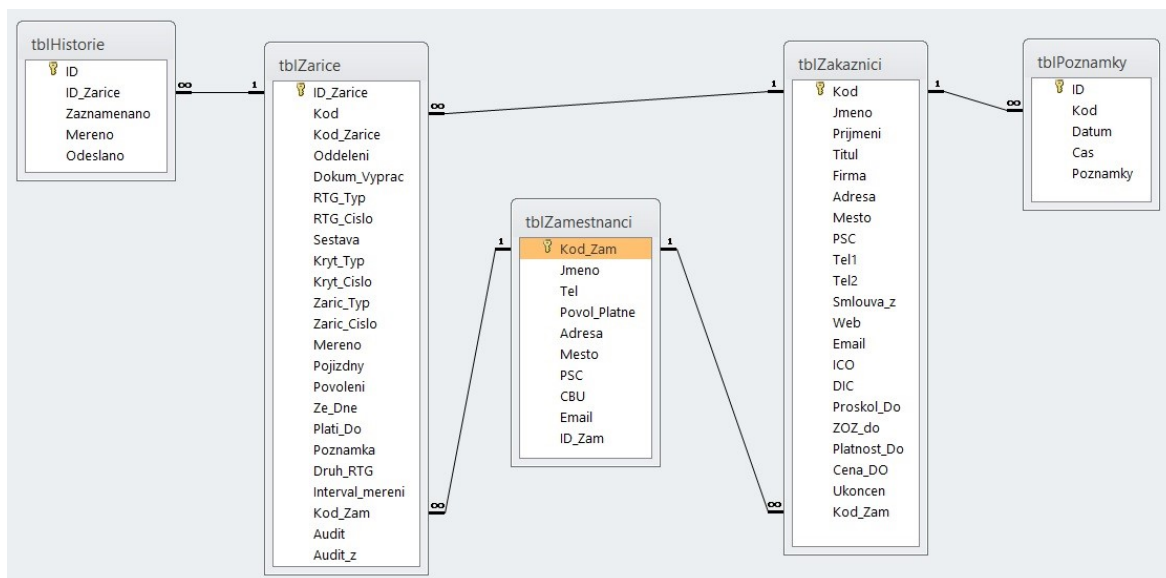
Název pole	Datový typ	Velikost pole	Nutno zadat	Vstupní maska	Indexovat
ID	Automatické číslo	Dlouhé celé číslo	-	-	Ano (bez duplicity)
ID_Zarice	Číslo	Dlouhé celé číslo	Ano	-	Ne
Zaznamenano	Datum a čas	-	Ano	-	Ne
Mereno	Datum a čas	-	Ano	-	Ne
Odeslano	Datum a čas	-	Ne	-	Ne

Tabulka 4.5: Návrh tabulky *tblHistorie*, zdroj: vlastní

4.3.2. Vztahy

Na obrázku 4.3 jsou již zobrazeny zrealizované tabulky a jejich vzájemné propojení pomocí vztahů v prostředí softwarové aplikace Microsoft Access 2010. Jedná se již o tzv. fyzický databázový model. Vztahy jsou vytvořeny na základě cizích a primárních klíčů, díky kterým jsou jednotlivé tabulky propojeny.

Kardinalita všech vztahů je 1:N, což znamená, že jeden zákazník může mít více záříčů, respektive poznámek, jeden zaměstnanec může provádět dohled více zákazníkům a může měřit více zákaznickových rentgenových záříčů a jeden záříč může mít více záznamů o historii jeho měření. Vztahu mezi tabulkami *tblZakaznici* a *tblZarice* jsou nastaveny vlastnosti zajišťující referenční integritu a kaskádové aktualizace souvisejících polí. Všechny ostatní vztahy mají krom těchto dvou vlastností nastavenou ještě vlastnost kaskádové odstranění souvisejících polí. Díky referenční integritě je zabráněno například přidání záznamu do tabulky *tblHistorie* k záříči, o kterém není veden záznam v tabulce *tblZarice*. Pomocí kaskádové aktualizace souvisejících polí je umožněno provádět změny v jedné tabulce, které se zároveň projeví i v tabulce přidružené. Kaskádové odstranění souvisejících polí znamená, že lze odstranit záznam z tabulky i v případě souvisejících záznamů v přidružené tabulce.



Obrázek 4.3: Návrh fyzického modelu, zdroj: vlastní

4.3.3. Formuláře

Následující kapitola je zaměřena na popis všech vytvořených formulářů sloužících pro obsluhu databázové aplikace. Pomocí formulářů je docíleno snadného ovládání tabulek, které zvládne i běžný uživatel bez znalostí principů tvorby databází.

Databázová aplikace k této bakalářské práci obsahuje několik formulářů sloužících k různým účelům. Pomocí některých z nich se záznamy v tabulkách upravují, případně přidávají nové. Další slouží k navigaci a zobrazování jednotlivých záznamů nebo k tisku sestav. Všem formulářům byl vytvořen jednoduchý grafický vzhled inspirovaný odstíny barev znaku varujícím před potencionálním radiačním nebezpečím, tj. žlutá a černá. Všechny formuláře obsahují ve svém záhlaví název firmy a její logo.

Formulář frmHlavni

Při spuštění databázové aplikace se jako první zobrazí právě tento formulář (Obrázek 4.4) sloužící k prvotní navigaci v databázové aplikaci. Ve formuláři je obsaženo několik tlačítek. Tlačítko *Zákazníci* slouží k otevření formuláře se zákazníky a jejich zářiči. Další tlačítko slouží k otevření formuláře se zaměstnanci zkušební laboratoře. Tlačítka *Měřeno v měsíci* a *Proškoleni v měsíci* slouží k otevření formulářů obsahující možnost výběru měsíce, ve kterém byli konkrétní zákazníci měřeni, respektive školeni. Pomocí tlačítka *Zákazníci dle města* je spuštěn parametrický dotaz, který uživatele vyzve k zadání konkrétního města, ze kterého chce udělat výpis zákazníků. Poslední tlačítko slouží k zobrazení sestavy všech zákazníků. V zápatí formuláře se nachází tlačítko sloužící k uzavření celé databázové aplikace, při jeho stisku je zobrazeno dialogové okno pro potvrzení uzavření.



Obrázek 4.4: Úvodní formulář, zdroj: vlastní

Formulář frmZakaznici

Na obrázku 4.5 je formulář sloužící k prohlížení zákazníků firmy z tabulky *tblZakaznici*. Informace o zákazníkovi a tlačítka sloužící k jejich obsluze jsou v levé části formuláře, napravo se nachází informace o rentgenovém zářiči konkrétního zákazníka. Jsou v něm obsažena navigační tlačítka, tlačítka pro otevírání sestav, přidávání, upravování a odstraňování záznamů, vyhledávací pole, apod.

Součástí formuláře jsou i podformuláře *frmPoznamky* zobrazující poznámky ke konkrétnímu zákazníkovi, *frmZarice*, pomocí kterého jsou zobrazeny zákaznickovy rentgenové přístroje a *frmPodformularHistorieMereni* vázaného k formuláři se zářiči zobrazující historii jejich měření. Je nutné zobrazovat veškeré tyto formuláře v jednom z důvodu potřeby přehledu o všech informacích spojených se zákazníkem. Veškerá pole v tom formuláři jsou uzamčená, aby nedocházelo k nevyžádaným úpravám, ale jsou zpřístupněna, díky čemuž je umožněno údaje z polí označovat a následně kopírovat.

Zákazníci RadEX Primar s.r.o. Nový zákazník Seznam zákazníků

AFN IČO 9845215 DIČ CZ9845215

MDDr. František Nowaczek

Dental FM s.r.o. Smlouva z 02.11.2014

Adresa Revoluční 3470 Telefon 1 737 625 854

Frýdek - Místek 738 03 Telefon 2 596 874 451

Email Web

Dohled

Smlouva DO od 02.11.2015 Dohlížejí EP1

Proškolen dne 02.11.2015 ZOZ do

Ukončen dohled Cena 1 500 Kč bez DPH

Procházení zákazníků Vyhledávání Tisk pasportu

Zobrazit vše Vyhledat

Upravit zákazníka Odstranit zákazníka Zákaznickovy RTG zář.

Datum přidání Čas Poznámka k zákazníkovi Nová

21.03.2016 16:49:15 zaslat výsledky testů Upravit Odstranit

Zářiče: Nový zářič

Kód zářiče AFN 01 Povolení 7341/2015

Oddělení Platné od 12.11.2015

Pojízdný Platné do 12.11.2025

RTG typ Thropy Irix 70 L Dokumentace 08.01.2016

v. č. CZ974312 Meřeno 02.11.2015

Audit 11/2016 Vyšel Měřil pracovník EP1

Sestava náradí

RTG Zářič Rentgenka Thropy 70 X Kryt/Komora DG024

v. č. XANK0245 v. č. 984325

Druh RTG Jednoduchý zdroj Interval 1 rok

Poznámka k zářiči

1 z 1 Upravit zářič Odstranit zářič

Souhrn historie

Historie měření: Nový záznam

Zaznamenáno Měřeno Protokol odeslán

21.03.2016 02.11.2014 18.11.2014

1 z 1 Upravit Odstranit

Zpět na úvod 2 z 5 pondělí 21. března 2016 18:58:38

Obrázek 4.5: Formulář se zákazníky, zdroj: vlastní

V záhlaví formuláře se kromě loga nachází dvě tlačítka, jedno slouží k otevření sestavy se všemi zákazníky a druhé otevření formuláře *frmNovy* (Obrázek 4.6), pomocí kterého je vytvářen záznam o novém zákazníkovi do tabulky *tblZakaznici*. V tomto formuláři je nutné vyplnit jedinečný kód zákazníka, který je zároveň primárním klíčem a příjmení nebo název firmy, jinak nebude možné vytvořit nový záznam a bude zobrazena chybová hláška. Ostatní pole jsou nepovinná, a to pro případ, že nejsou známy veškeré údaje o zákazníkovi. Součástí i tohoto formuláře je tlačítko pro zobrazení sestavy všech zákazníků. Tlačítko *Vytvořit* slouží k zapsání uvedených informací do tabulky *tblZakaznici*. V zápatí se nachází tlačítko na uzavření tohoto přidávacího formuláře.

Pod údaji o zákazníkovi se v tomto formuláři nachází navigační tlačítka pro přechod mezi jednotlivými zákazníky, vedle kterých je pole pro vyhledávání zákazníka. Tlačítko *Vyhledat* slouží k vyhledání zákazníka na základě zadaného textu ve vyhledávacím poli a proběhne vyhledání odpovídajících záznamů. Oblastí vyhledávání jsou údaje kód, příjmení, firma, adresa, město a IČO v tabulce *tblZakaznici*. V případě hledání neexistujícího záznamu je zobrazena chybová hláška. Vedle vyhledávacího tlačítka se nachází tlačítko pro zobrazení všech záznamů.

Pomocí tlačítka *Upravit zákazníka* je otevřen formulář *frmUpravit* (Obrázek 4.6) sloužící k úpravě údajů konkrétního zákazníka. Jedná se o obdobný formulář jako formulář pro přidání nového zákazníka s tím rozdílem, že je upravován již existující záznam. Lze v něm upravovat veškeré údaje o zákazníkovi kromě jeho kódu, který je primárním klíčem. K odstranění záznamu o zákazníkovi slouží tlačítko nacházející se vedle tlačítka k upravování. Při upravování i odstraňování je vyžadováno potvrzení. Napravo od tlačítka sloužícího k odstranění zákazníka se nachází tlačítko pro tisk sestavy zobrazující všechny zákaznickovy rentgenové zářiče.

The image shows two side-by-side screenshots of a software application window titled 'RadEX Primar s.r.o.'. The left window is 'Nový zákazník' (New customer) and the right window is 'Upravit zákazníka' (Edit customer). Both windows contain a form with various input fields for customer data.

Nový zákazník (Left Window):

- Kód zákazníka:** IČO (input), DIČ (input)
- Titul:** (input), **Jméno:** (input), **Příjmení:** (input)
- Firma:** (input), **Smlouva z:** (input)
- Adresa:** Ulice a č. p. (input), Město (input), PSČ (input)
- Telefon 1:** (input), **Telefon 2:** (input), **Web:** (input), **E-mail:** (input)
- Dohled:** Smlouva DO od (input), Proškolen dne (input), Ukončen dohled (input), ZOZ do (input), Dohlížející (dropdown), Cena (input) bez DPH
- Buttons:** Vytvořit, Seznam zákazníků, Zpět na zákazníky

Upravit zákazníka (Right Window):

- Kód zákazníka:** AFN (input), IČO 9845215 (input), DIČ C29845215 (input)
- Titul:** MDDr. (input), **Jméno:** František (input), **Příjmení:** Nowaczek (input)
- Firma:** Dental FM s.r.o. (input), **Smlouva z:** 02.11.2014 (input)
- Adresa:** Ulice a č. p. (input), Revoluční 3470 (input), Město (input), PSČ (input)
- Telefon 1:** 737 625 854 (input), **Telefon 2:** 596 874 451 (input), **Web:** (input), **E-mail:** (input)
- Dohled:** Smlouva DO od 02.11.2015 (input), Proškolen dne 02.11.2015 (input), Ukončen dohled (input), ZOZ do (input), Dohlížející EP1 (dropdown), Cena 1 500 Kč (input) bez DPH
- Buttons:** Upravit, Zpět na zákazníky

Obrázek 4.6: Formuláře k přidávání a upravování záznamu o zákazníkovi, zdroj: vlastní

Tlačítko *Tisk pasportu* slouží k exportování konkrétních polí konkrétního zákazníka do externí přesně strukturované šablony aplikace Excel. Jedná se o průvodní list zákazníka potřebný pro interní úkony zkušební laboratoře. Tato funkčnost výrazně urychluje vyplňování jednotlivých pasportů. Pro znázornění je v příloze č. 1 zobrazena část pasportu.

V zápatí formuláře je na levé straně tlačítko k uzavření tohoto formuláře, uprostřed se nachází počítadlo záznamů zobrazující aktuální záznam z celkového počtu záznamů a napravo se zobrazuje aktuální datum a čas.

Formulář frmPoznamky

V levé spodní části formuláře *frmZakaznici* se nachází podformulář s poznámkami k zákazníkovi, které jsou řazeny od nejnovějších po nejstarší ve formě nekonečného formuláře. Poznámky lze přidávat, upravovat a odstraňovat pomocí příslušných tlačítek. Tlačítka *Nová* a *Upravit* slouží k otevření formuláře *frmNovaPozn* (Obrázek 4.7), pomocí kterého se vytvoří k zákazníkovi nová poznámka a vloží se do tabulky *tblPoznamky*, respektive formuláře *frmUpravPozn* (obrázek 4.7), který slouží k upravení již existující poznámky. Při jejich otevření je automaticky přiděleno aktuální datum a čas značící dobu přidání poznámky. Součástí obou těchto formulářů je v zápatí tlačítko *Zpět*, které je uzavře. Při upravování a odstraňování poznámky je opět vyžadováno potvrzení formou dialogového okna.

Datum přidání	Čas	Poznámka k zákazníkovi
21.03.2016	19:11:21	

Kód zákazníka: AFN

Přidat

Zpět

Obrázek 4.7: Formuláře k přidávání a upravování poznámek, zdroj: vlastní

Formulář frmZarice

Ve formuláři *frmZakaznici* se rovněž nachází podformulář s rentgenovými zářiči daného zákazníka a je situován v jeho pravé části. Jsou v něm obsaženy veškeré potřebné údaje o jednotlivých zářičích patřící konkrétnímu zákazníkovi. Záznamy jsou řazeny podle kódu zářiče.

Tlačítko *Nový zářič* slouží k otevření formuláře *frmNovyZaric* (Obrázek 4.8), díky němuž je umožněno přidávat do tabulky *tblZarice* nové záznamy. Při otevření formuláře je do pole *Kód zářiče* automaticky vložen kód zákazníka a uživatel by posléze měl doplnit

do tohoto pole číslo značící pořadí aktuálního zákaznickova záříče. Jedná se o jediné povinné pole, které musí být vyplněno a zároveň musí být jedinečné. Ostatní pole mohou být doplněna dodatečně.

Mezi záznamy o záříčích lze přecházet pomocí navigačních tlačítek situovaných ve spodní části. Vedle nich se nachází počítadlo záznamů a tlačítka sloužící k úpravě a odstranění záznamu o záříči. Při stisku tlačítka *Upravit záříč* je otevřen formulář *frmUpravZaric* (Obrázek 4.8), ve kterém lze údaje o záříči upravit. V zápatí formulářů pro úpravu i pro vytvoření nového záznamu o záříči je tlačítko pro jejich uzavření.

Obrázek 4.8: Formuláře k přidávání a upravování záznamu o záříči, zdroj: vlastní

Posledním tlačítkem v tomto formuláři je tlačítko *Souhrn historie*, pomocí kterého je zobrazena sestava s výpisem celé historie měření daného rentgenového zařízení.

Formulář *frmPodformularHistorieMereni*

Posledním formulářem zobrazujícím se na formuláři *frmZakaznici* je podformulář obsahující historii měření daného rentgenového záříče. Jedná se o podformulář formuláře *frmZarice*, který byl popsán výše. Tlačítkem *Nový záznam* je otevřen formulář *frmNovyZaznamHistorie* (Obrázek 4.9), pomocí kterého je vkládán nový záznam o historii měření záříče do tabulky *tblHistorie*. Automaticky je přidáno datum, kdy byl záznam přidán a datum, kdy byl naposled zákazníkům záříč měřen.

Součástí formuláře jsou tlačítka pro procházení jednotlivých záznamů, vedle kterých se nachází počítadlo těchto záznamů. Dále jsou zde tlačítka pro úpravu a odstranění záznamu. Formulář pro upravování záznamu o historii měření je na obrázku 4.9 a jedná se o téměř identický formulář, jako byl ten pro přidávání nového záznamu s tím rozdílem, že tímto formulářem se upravují již existující data v tabulce. U obou formulářů se v zápatí nachází tlačítko pro jejich uzavření. Při upravování i odstraňování záznamu je vyžadováno potvrzení.

Zaznamenáno	Měřeno	Protokol odeslán
21.03.2016	02.11.2015	

Kód záříče: AFN 01

Vytvořit záznam v historii

Zpět

Obrázek 4.9: Formuláře k přidávání a upravování záznamu o historii záříče, zdroj: vlastní

Formulář frmZamestnanci

Na základě stisknutí tlačítka *Zaměstnanci* na úvodním formuláři je otevřen formulář (Obrázek 4.10) s technickými pracovníky zkušební laboratoře a se zdrojem dat v tabulce *tblZamestnanci*, kteří provádí měření a dohled u zákazníků. Obdobně jako u předchozích formulářů lze přidávat nové záznamy, upravovat je, respektive mazat. Stejně jako u ostatních formulářů jsou i zde všechna pole uzamčena a zpřístupněna.

RadEX Primar s.r.o. Nový zaměstnanec

Tomáš Nový TN1 Telefon 605 987 789

Adresa Josefa Skupy 1641 E-mail tomas.novy@radex.cz

Ostrava 708 00 ČBU 000000-2154785425/0800

Povolení platné do 30.06.2018

Měřeno v měsíci Měřeno celkem

Upravit zaměstnance Odstranit zaměstnance Dohled

Zpět na úvod 2 z 2

Obrázek 4.10: Formulář se zaměstnanci, zdroj: vlastní

Tlačítko v záhlaví slouží k otevření formuláře *frmNovyZam* (Obrázek 4.11), pomocí kterého se vkládají nové záznamy do tabulky *tblZamestnanci*. Povinnými údaji jsou jméno zaměstnance a jeho jedinečný kód sloužící jako primární klíč. Ostatní údaje je možno doplnit dodatečně.

Pod údaji o zaměstnanci se ve formuláři krom navigačních tlačítek nachází mnohá další. Pomocí tlačítka *Upravit zaměstnance* je otevřen formulář *frmUpravZam* (Obrázek 4.11) sloužící k upravování údajů o zaměstnanci, který je téměř totožný s formulářem přidávacím, akorát se pomocí něj upravují již existující záznamy v tabulce. V těsné blízkosti se nachází tlačítko pro odstranění právě vybraného záznamu. Při úpravě i odstraňování záznamu o zaměstnanci je opět vyžadováno potvrzení.

Obrázek 4.11: Formuláře k přidávání a upravování záznamu o zaměstnanci, zdroj: vlastní

Tlačítko *Měřeno celkem* slouží k zobrazení sestavy souhrnu celkového měření daného zaměstnance za poslední rok a tlačítko *Dohled* k zobrazení sestavy souhrnu všech zákazníků, kterým dělá daný zaměstnanec dohled.

Stisknutí tlačítka *Měřeno v měsíci* způsobí otevření formuláře *frmZamMerilMesic* (Obrázek 4.12), pomocí kterého je možné zobrazit sestavy měření daného zaměstnance za daný měsíc.

Obrázek 4.12: Formulář s výběrem měsíčního měření zaměstnance, zdroj: vlastní

Formuláře frmMereno a frmProskoleno

Na základě stisknutí tlačítek *Měřeno v měsíci* a *Proškolení v měsíci* na úvodním formuláři jsou otevřeny formuláře *frmMereno*, respektive *frmProskoleno*, které lze vidět na obrázku 4.13. Formuláře jsou téměř identické s tím rozdílem, že první z nich slouží po stisku konkrétního měsíce k zobrazení sestavy souhrnu záříčů měřených v onom měsíci a druhý z nich k zobrazení sestavy souhrnu zákazníků proškolených v daném měsíci. Oba formuláře mají ve svém zápatí tlačítko pro jejich uzavření.

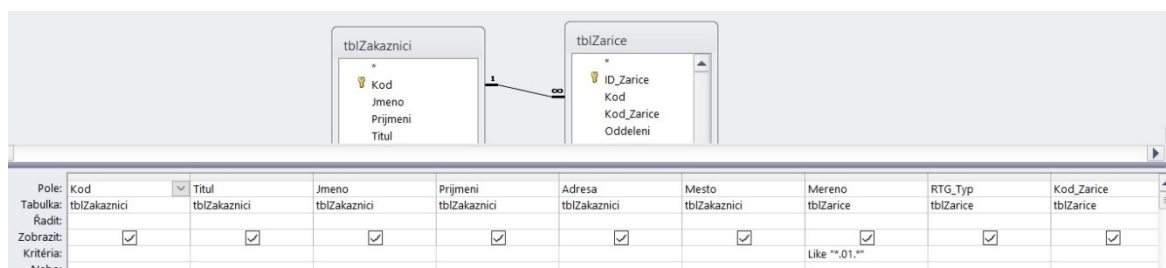


Obrázek 4.13: Formuláře s výběrem měsíčního měření a proškolení zákazníků, zdroj: vlastní

4.3.4. Dotazy

V databázové aplikaci je vytvořeno velké množství dotazů. Byly využity především výběrové dotazy, na základě kterých jsou následně generovány sestavy z tabulek.

Byly vytvořeny dotazy pro zjištění souhrnu proškolených zákazníků a souhrnu měřených záříčů za daný měsíc. Tyto dotazy byly vytvořeny jednotlivě pro každý měsíc. Příklad takového dotazu je dotaz *dtzLeden* na obrázku 4.14, který slouží pro zjištění souhrnu měřených záříčů za měsíc leden, kde je nutností zobrazit i údaje o zákazníkovi.

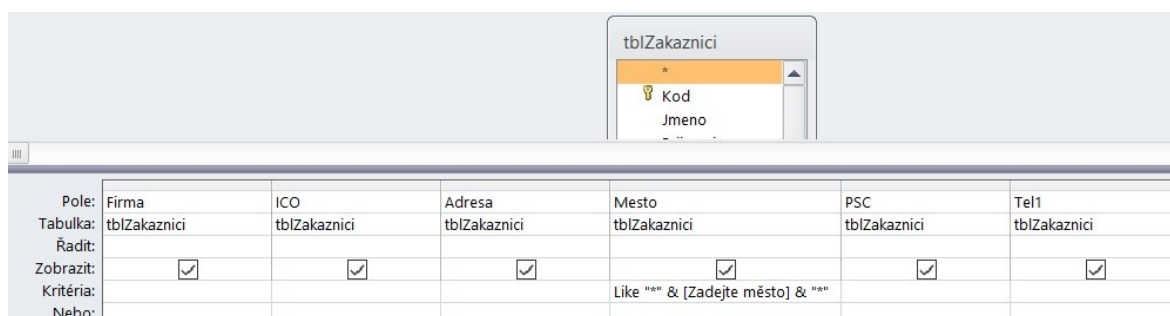


Pole:	Kod	Titul	Jmeno	Prijmeni	Adresa	Mesto	Mereno	RTG_Typ	Kod_Zarice
Tabulka:	tblZakaznici	tblZakaznici	tblZakaznici	tblZakaznici	tblZakaznici	tblZakaznici	tblZarice	tblZarice	tblZarice
Radit:									
Zobrazit:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kritéria:							Like **01.**		
Nebo:									

Obrázek 4.14: Návrhové zobrazení dotazu *dtzLeden*, zdroj: vlastní

Dotazy na stejném principu byly vytvořeny i pro zjištění měsíčního měření každého zaměstnance. U každého záříče je možné zobrazit celkovou historii jeho měření, čehož je také docíleno pomocí dotazu.

Pro zobrazení zákazníků z určitého města slouží parametrický dotaz *dtzZakMesto* (Obrázek 4.15), jehož výsledek není přímo specifikován v návrhu, ale je určen až po jeho spuštění na základě výzvy a následného vepsání požadovaného názvu města.



Pole:	Firma	ICO	Adresa	Mesto	PSC	Tel1
Tabulka:	tblZakaznici	tblZakaznici	tblZakaznici	tblZakaznici	tblZakaznici	tblZakaznici
Řadit:						
Zobrazit:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kritéria:				Like *** & [Zadejte město] & ***		
Nebo:						

Obrázek 4.15: Návrhové zobrazení dotazu *dtzZakMesto*, zdroj: vlastní

Dalším typem dotazů, který byly při tvorbě databázové aplikace použity, jsou dotazy napsané v programovacím jazyku VBA. Slouží k přidávání, upravování nebo odstraňování záznamů z tabulky prostřednictvím formulářů. Konkrétně byly použity dotazy přidávací, aktualizací a odstraňovací. Na obrázku 4.16 je možné vidět část všech tří druhů použitých dotazů tvořených pro práci se záznamy v tabulce se zaměstnanci. První z nich je přidávací s klíčovým slovem INSERT, druhý v pořadí zobrazený na obrázku je aktualizací dotaz s klíčovým slovem UPDATE a poslední z nich je dotaz odstraňovací s příkazem DELETE.

```
prik.CommandText = "INSERT INTO tblZamestnanci(Kod_Zam, Jmeno, Tel, Povol_Platne, Adresa, Mesto, PSC, CBU, Email) " & _
"VALUES(?, ?, ?, ?, ?, ?, ?, ?, ?)"

prik.CommandText = "UPDATE tblZamestnanci " & _
"SET Jmeno=?, Tel=?, Povol_Platne=?, Adresa=?, Mesto=?, PSC=?, CBU=?, Email=?" & _
"WHERE Kod_Zam=?"

prik.CommandText = "DELETE FROM tblZamestnanci WHERE Kod_Zam=?"
```

Obrázek 4.16: Části dotazů pro práci se záznamy tabulky *tblZamestnanci*, zdroj: vlastní

4.3.5. Sestavy

Na základě výběrových dotazů bylo vytvořeno několik sestav sloužících k zobrazení určitých dat z tabulek. Všem sestavám bylo nastaveno podobné formátování, aby působily uceleným dojmem.

Databázová aplikace obsahuje sestavu zobrazující celkový výpis všech záznamů z tabulky se zákazníky, dále sestavy zobrazující konkrétní souhrny na základě výběru daného měsíce, jako je měření v měsíci, proškolení v měsíci, nebo zaměstnancovo měření

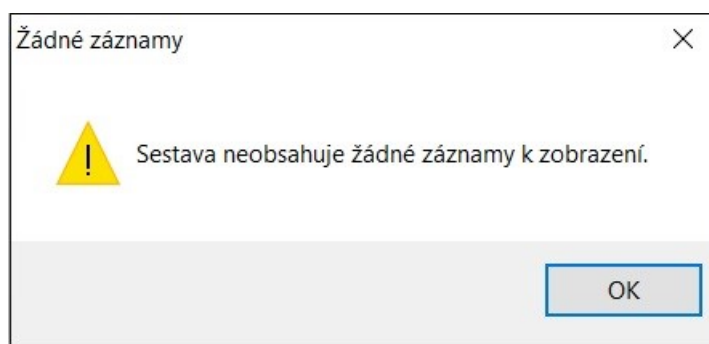
v měsíci. Příkladem může být sestava v příloze č. 2 zobrazující souhrn zákazníků a jejich rentgenových zařízení měřených v měsíci listopadu.

Dalším typem sestav, objevující se v tvořené databázové aplikaci, jsou sestavy zobrazující data z tabulek, které souvisejí s konkrétním záznamem. Jedná se například o zobrazení souhrnu všech rentgenových záříčů patřících konkrétnímu zákazníkovi (Příloha č. 3), nebo zobrazení celkové historie daného záříče, apod.

Sestava zobrazující souhrn zákazníků z konkrétního města je tvořena na základě parametrického dotazu, ve kterém uživatel napíše požadované město.

Přístup k jednotlivým sestavám je již popsán dříve v této kapitole v části zabývající se formuláři.

V případě, že by nebyly žádné záznamy k zobrazení, je vyhozena chybová hláška (Obrázek 4.17) a prázdná sestava není zobrazena.



Obrázek 4.17: Chybová hláška při zobrazování prázdných sestav, zdroj: vlastní

4.3.6. Dokončení aplikace

S databázovou aplikací sice bude pracovat pouze jeden zaměstnanec zkušební laboratoře a aplikace bude pouze na jednom počítači, ale i tak je nutné aplikaci nějakým způsobem zabezpečit. Navíc i přes to, že aktuálně to sice není naplánováno a bráno v potaz, tak je možné, že v budoucnu bude databáze nesdílena mezi více zaměstnanců, kteří s ní budou moci pracovat.

V databázové aplikaci byly proto skryty tlačítka a většina karet, aby nebylo možné samovolně aplikaci upravovat, případně nějaký objekt omylem nenávratně smazat. Běžnému uživateli se při spuštění aplikace zobrazí vyjma úvodního formuláře pouze karta *Domů* se základními tlačítky. Bude-li potřeba databázovou aplikaci upravit, stačí ji spustit se stisknutou klávesou Shift a zobrazí se opět všechny karty a možnosti, což se však běžnému uživateli nedoporučuje.

4.4. Implementace aplikace do firmy

Databázová aplikace bude zkušební laboratoři předána na paměťovém médiu. Zaměstnanec zkušební laboratoře vloží aplikaci do svého počítače, který je hardwarově dostačující a je na něm nainstalován kancelářský balíček Microsoft Office 2010. Reálná data budou do aplikace vložena firmou ve vlastní režii. Databázová aplikace může být na základě dalších požadavků průběžně upravována a můžou v ní být odstraňovány případné zjištěné nedostatky.

5. Závěr

Cílem této bakalářské práce bylo navrhnout a reálně vytvořit databázovou aplikaci klientů pro zkušební laboratoř RadEX Primar s.r.o. zabývající se prováděním zkoušek diagnostických zdravotnických rentgenových přístrojů. S aplikací bude pracovat pouze jeden zaměstnanec zabývající se vyjma jiných věcí domlouváním a korigováním provádění zkoušek u klientů, který bude vycházet právě z údajů v databázi a bude je upravovat do aktuální podoby.

V první části bakalářské práce byly popsány teoretické pojmy a východiska tvorby databází. Byly zde popsány základy nutné k tvorbě databázové aplikace, jako jsou elementární pojmy, které souvisí s databází, pojem datové modelování složené z konceptuálního, logického a fyzického modelu, popisující návrh databáze a v neposlední řadě popis a seznámení s programovým vybavením Microsoft Access 2010, ve kterém byl tvořen praktický výstup této bakalářské práce.

Další kapitola byla zaměřena na společnost, pro kterou byla aplikace tvořena. Byly zde sepsány požadavky na databázovou aplikaci, které byly analyzovány na základě několika schůzek s jednatelem. Jelikož původní stav ve firmě nedostačoval aktuálním požadavkům firmy, bylo nutné splnit veškeré požadavky na nově tvořenou databázovou aplikaci.

Následovala stěžejní část bakalářské práce zabývající se jejím praktickým řešením. Na základě požadavků byly postupně vytvořeny konceptuální a logický model pomocí entitně-relačního diagramu. Z těchto modelů byla posléze provedena praktická realizace tabulek a jejich vzájemné propojení v prostředí programu Microsoft Access 2010. Následovalo vytvoření všech obslužných formulářů, dotazů a sestav. To vše bylo v této části popsáno, včetně popisu celkové funkčnosti databázové aplikace.

Splněny byly veškeré požadavky zadavatele a tudíž i celkový cíl bakalářské práce a vytvořenou databázovou aplikaci je možno začít používat.

Zadavatel během průběžných schůzek jevil spokojenost nad tvořenou aplikací a jeho prvotní dojem s výslednou aplikací byl dobrý. Je samozřejmě možné, že se během praktického užívání vyskytnou případné nedostatky, které bude potřeba dodatečně eliminovat, aby bylo dosaženo maximální užitečnosti a spokojenosti s databázovou aplikací.

Seznam použité literatury

Literatura:

CONOLLY, T., C. BEGG a R. HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.

HERNANDEZ, Michal, J. *Návrh databází*. Praha: Grada Publishing, 2006. ISBN 80-247-0900-7.

KALUŽA, Jindřich a Ludmila KALUŽOVÁ. *Modelování dat v informačních systémech*. Praha: Ekopress, 2012. ISBN 978-80-86929-81-1.

KROENKE, David a David J. AUER. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.

KRUCZEK, Aleš. *Microsoft Access 2010: podrobná uživatelská příručka*. Brno: Computer Press, 2010. ISBN 978-80-251-3289-0.

OPPEL, Andrew J. *Databáze bez předchozích znalostí*. Brno: Computer Press, 2006. ISBN 80-251-1199-7.

OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Brno: Computer Press, 2008. ISBN 978-80-251-1707-1.

PÍSEK, Slavoj. *Access 2010: podrobný průvodce*. Praha: Grada Publishing, 2011. ISBN 978-80-247-3653-2.

SHEPHERD, Richard. *Access VBA: výukový průvodce*. Brno: Computer Press, 2012. ISBN 978-80-251-3686-7.

ŠIMONOVÁ, Stanislava a Jan PANUŠ. *Databázové systémy I*. Pardubice: Univerzita Pardubice, 2007. ISBN 978-80-7194-988-6.

Internetové zdroje:

KULHAN, Jakub. *Normalizace relačních databází* [online]. 23. 7. 2008.

[cit. 2015-02-11]. Dostupné z: <http://programujte.com/clanek/2008071900-normalizace-relacnich-databazi/>.

NOVÁK, Vítězslav. *Přednášky předmětu: Základy databází* [online]. 2014. [cit. 2016-02-10]. Dostupné z: <http://lms.vsb.cz/course/view.php?id=8933>

Seznam zkratek

1NF – první normální forma

2NF – druhá normální forma

3NF – třetí normální forma

4NF – čtvrtá normální forma

5NF – pátá normální forma

BCNF – Boyce Coddova normální forma

BÚ – bankovní účet

CK – cizí klíč

DBMS – Database Management System

DCL – Data Control Language

DDL – Data Definition Language

DIČ – Daňové identifikační číslo

DML – Data Manipulation Language

DO – dohlízející osoba

ER – entitně-relační

ERD – entitně-relační diagram

IBM – International Business Machines Corporation

IČO – identifikační číslo osoby

IMS – Information Management System

PHP – Hypertext Preprocessor

PK – primární klíč

PSČ – poštovní směrovací číslo

RTG – rentgen, rentgenový

SQL – Structured Query Language

SŘBD – systém řízení báze dat

SÚJB – státní úřad pro jadernou bezpečnost

TCL – Transaction Control Language

VB – Visual Basic

VBA – Visual Basic for Application

VBE – Visual Basic Editor

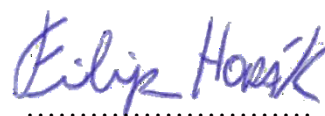
ZOZ – zkouška odborné způsobilosti

Prohlášení o využití výsledků bakalářské práce

Prohlašuji, že

- jsem byl seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. – autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo;
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně, ke své vnitřní potřebě, bakalářskou práci užít (§ 35 odst. 3);
- souhlasím s tím, že bakalářská práce bude v elektronické podobě archivována v Ústřední knihovně VŠB-TUO a jeden výtisk bude uložen u vedoucího bakalářské práce. Souhlasím s tím, že bibliografické údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO;
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona;
- bylo sjednáno, že užít své dílo, bakalářskou práci, nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).

V Ostravě dne 6. 5. 2016



Filip Horák

Seznam příloh

Příloha č. 1: Část vzorového pasportu konkrétního zákazníka


Příloha č. 2: Ukázková sestava zářičů měřených v listopadu

Příloha č. 3: Ukázková sestava zákaznickových zářičů

Příloha č. 4: CD s databázovou aplikací

Přílohy

Příloha č. 1

 RadEX Primar s.r.o.	Průvodní list	identifikace zakazníka AFN												
Stručná charakteristika firmy: RadEX Primar s.r.o. je Státním úřadem pro jadernou bezpečnost pověřenou specializovanou zkušební laboratoří. Předmětem zkoušek, které laboratoř provádí, jsou diagnostické zdravotnické rentgenové přístroje a rentgenová pracoviště jako celky (více na www.radex.cz).														
Specifikace požadavky: Telefonická výzva k provedení ZDS														
Termín měření (datum, čas):	Místo měření (adresa, kontaktní osoba)	Počet zářídí:												
<table border="1"><tr><td>Informace o zákazníkovi:</td><td>Firma: Dental FM s.r.o.</td></tr><tr><td>IČO: 9845215</td><td>Ředitel: MDDr. František Nowaczek</td></tr><tr><td>DIČ: CZ9845215</td><td>Adresa: Revoluční 3470</td></tr><tr><td>Tel. 1: 737625854</td><td>Město: Frýdek - Místek</td></tr><tr><td>Tel. 2: 596874451</td><td>Kraj-RC-SÚJB:</td></tr><tr><td>e-mail:</td><td>fakturovat na:</td></tr></table>			Informace o zákazníkovi:	Firma: Dental FM s.r.o.	IČO: 9845215	Ředitel: MDDr. František Nowaczek	DIČ: CZ9845215	Adresa: Revoluční 3470	Tel. 1: 737625854	Město: Frýdek - Místek	Tel. 2: 596874451	Kraj-RC-SÚJB:	e-mail:	fakturovat na:
Informace o zákazníkovi:	Firma: Dental FM s.r.o.													
IČO: 9845215	Ředitel: MDDr. František Nowaczek													
DIČ: CZ9845215	Adresa: Revoluční 3470													
Tel. 1: 737625854	Město: Frýdek - Místek													
Tel. 2: 596874451	Kraj-RC-SÚJB:													
e-mail:	fakturovat na:													

Příloha č. 2

Měřeno v listopadu

Kód	Měřeno	Titul	Jméno	Příjmení	Adresa	Město	RTG Typ	Kód RTG	Audit
STH	02.11.2015	MDDr.	Filip	Vejvoda	Hoblíkova 580	Nový Jičín	Thropy Elit	STH 01	<input type="checkbox"/>
AFN	02.11.2015	MDDr.	František	Nowaczek	Revoluční 3470	Frýdek - Místek	Thropy Irix	AFN 01 11/2016	<input checked="" type="checkbox"/>
FNO	04.11.2015		Jan	Petrovský	17. listopadu 19	Ostrava	Mindent 80	FNO 01	<input type="checkbox"/>

Zákazníkovy RTG zářiče

Kód	Titul	Příjmení	Firma	IČO	Město	Kód Zářiče	RTG Typ	Měřeno
FNO		Petrovský	Fakultní nemocnice	843 989	Ostrava	FNO 01	Mindent 80 D	04.11.2015
						FNO 02	TMS 152	07.12.2015
						FNO 03	VMX plus	11.02.2016